Host Programmer Interface Specification for the NetXtreme® Family of Highly Integrated Media Access Controllers

5300 California Avenue • Irvine, CA 92617 • Phone: 949-926-5000 • Fax: 949-926-5203

Revision History

Revision	*Date*	*Description*
57XX-PG105-R	01/29/08	**Updated:** • Bus Interfaces in Table 1: "Family Features," on page 2 • "NVRAM Configuration" on page 88 • Step 4 in "Initialization Procedure" on page 146 • Added notes to "Indirect Register Access" on page 186 and "Indirect Memory Access" on page 188 **Removed:** • "NVRAM Control"
57XX-PG104-R	07/13/06	**Updated:** • ASF support parameters for Dual MAC Transceivers in Table 1: "Family Features," on page 2. • BCM5703C and BCM5703S MAC transceiver rows in Table 2: "Family Revision Levels," on page 4. • "Pseudocode to Access NVRAM in Auto Access Mode" on page 108. • Unique identifier information for BCM5715C and BCM5715S in Table 144: "Device ID Register (Offset 0x02)," on page 330. • Bit 9 information in Table 184: "Miscellaneous Host Control Register (Offset 0x68)," on page 355. • Updated Bits 21-31 in Table 191: "PCI Clock Control Register Definition for BCM5714, and BCM5715 Devices," on page 369. • Bit 5 information Table 295: "Hardware Auto-Negotiation Control Register (Offset 0x5B0, BCM5704S Only)," on page 438. • Table 311: "Statistics Registers," on page 445. • Bit 18 information in Table 349: "Receive List Placement Statistics Enable Mask Register (Offset 0x2018)," on page 473. • Bits17-16 in Table 423: "Read DMA Mode Register (Offset 0x4800)," on page 519. • Bit 10 information Table 426: "Write DMA Mode Register (Offset 0x4C00)," on page 523. • Bits 6-8 in Table 450: "Mode Control Register (Offset 0x6800)," on page 545. • Bits 4-6 in Table 452: "Miscellaneous Local Control Register (Offset 0x6808)," on page 551. • Descriptive text in Table 524: "NVM Write Register (Offset 0x7008)," on page 603. • Descriptive text in Table 525: "NVM Address Register (Offset 0x700C)," on page 603. • Descriptive text in Table 526: "NVM Read Register (Offset 0x7010)," on page 604. • Bit 19, 15, 14, and 13 descriptive text in Table 530: "Software Arbitration Register (Offset 0x7020)," on page 607. • Access information in Table 627: "PHY Receive Error Counter (Offset 0x7E20)," on page 652. • Table 628: "PHY Receive Framing Error Counter (Offset 0x7E24)," on page 652. • Bit 1 and 2 information in Table 657: "PHY Extended Control Register (PHY_Addr = 0x1, Reg_Addr = 10h)," on page 678.

Revision	*Date*	*Description*
		• Bit 14 and 15 information in Table 658: “PHY Extended Status Register (PHY_Addr = 0x1, Reg_Addr = 11h),” on page 681. • Bits 1-5 in Table 666: “Expansion Register 01h: Expansion Interrupt Status,” on page 686. • Bit 3-5 information in Table 675: “18h: Auxiliary Control Register (Shadow Register Selector = "000"; BCM5714 and BCM5715 only),” on page 695. • Bits 8:7 and bit 3 information in Table 676: “18h: 10Base-T Register (Shadow Register Selector = "001"; BCM5714 and BCM5715 only),” on page 696. • Bit 5 information in Table 678: “18h: Miscellaneous Test Register 1 (Shadow Register Selector = "100"; BCM5714 and BCM5715 only),” on page 699. • Bit 14:12 and bit 10:5 information Table 679: “18h: Miscellaneous Test Register 2 (Shadow Register Selector = "101"; BCM5714 and BCM5715 only),” on page 700. • Bit 15 information in Table 682: “Auxiliary Control Register (PHY_Addr = 0x1, Reg_Addr = 18h, Shadow = 001, 10BASE-T),” on page 704. • Bit 6 information in Table 695: “Auto Power-Down Register (Address 1Ch, Shadow Value 01010),” on page 724. • Bit 7:4 and bit 3:0 information in Table 709: “LED GPIO Control/Status Register (Address 1Ch, Shadow Value 01111),” on page 741. • Bit 8:3 information in Table 710: “Autodetect SGMII/Media Converter Register (Address 1Ch, Shadow Value 11000),” on page 742. • Bits 5-8 and bit 3 information in Table 713: “Auxiliary 1000BASE-X Status Register (Address 1Ch, Shadow Value 11100),” on page 746. • Bit 7:6 information in Table 714: “Miscellaneous 1000BASE-X Status Register (Address 1Ch, Shadow Value 11101),” on page 749. • Bit 9:8 and bit 5 information in Table 715: “Autodetect Medium Register (Address 1Ch, Shadow Value 11110),” on page 751. • Bit 3 information Table 716: “Mode Control Register (Address 1Ch, Shadow Value 11111),” on page 753. • Bit 14:13 and bit 12 information in Table 717: “HCD Status Register (PHY_Addr = 0x1, Reg_Addr = 1Dh, Bit 15 = 1),” on page 755. • Note to Table 26: “Boot Strap Region,” on page 91. • Offset 9C, 9E, AC and B4 values to Table 28: “Manufacturing Information Region,” on page 93. • Bit 28 and Bit 29 to Table 28: “Manufacturing Information Region,” on page 93. • Bits 7-31 to Offset DC Table 28: “Manufacturing Information Region,” on page 93. • Procedural steps to “Initialization Procedure” on page 174. • Note in “Register Definitions” on page 327. • Descriptive note to Table 216: “Device Control Register (Offset 0xD8),” on page 386. • “00-0FH 1000BT/100BTX/10BT Registers” on page 658. • “Expansion Registers (BCM5705, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 Only)” on page 685. • “Auxiliary Control Register (BCM5714 And BCM5715 Devices Only)” on page 695

Revision	*Date*	*Description*
57XX-PG104-R	07/13/06	**Added:** • Table 675: "18h: Auxiliary Control Register (Shadow Register Selector = "000"; BCM5714 and BCM5715 only)," on page 695. • Table 676: "18h: 10Base-T Register (Shadow Register Selector = "001"; BCM5714 and BCM5715 only)," on page 696. • Table 677: "18h: Power/MII Control Register (Shadow Register Selector = "010"; BCM5714 and BCM5715 only)," on page 698. • Table 678: "18h: Miscellaneous Test Register 1 (Shadow Register Selector = "100"; BCM5714 and BCM5715 only)," on page 699. • Table 679: "18h: Miscellaneous Test Register 2 (Shadow Register Selector = "101"; BCM5714 and BCM5715 only)," on page 700. • Table 680: "18h: Miscellaneous Control Register (Shadow Register Selector = "111")," on page 700. • Table 690: "Spare Control Register 1 (Address 1Ch, Enable by Register 1Ch Bits[14:10] = 00010)," on page 718. • "Misc Shadow Registers (PHY_ADDR = 0X1, REG_ADDR = 1CH; BCM5702, BCM5703, and BCM5704 Only)" on page 718. • Table 691: "Spare Control Register 2 (Address 1Ch, Shadow Value 00100)," on page 719. • "Misc Shadow Registers (PHY_ADDR = 0X1, REG_ADDR = 1CH; BCM5705, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 Only)" on page 725. • Table 699: "Clock Alignment Control Register (Shadow Register Selector = "00011")," on page 728. • Table 702: "Spare Control 3 Register (Address 1Ch, Shadow Value 00101)," on page 730. • Table 703: "1Ch: Spare Control 3 Register (Shadow Register Selector = "00100")," on page 732. • Table 721: "1Fh: Test Register 2," on page 758.

Revision	*Date*	*Description*
57XX-PG103-R	08/18/05	**Updated:** • Table 2: “Family Revision Levels,” on page 5. • Figure 15: “Typical BCM5714C-Based LOM Design Block Diagram,” on page 39. • Figure 16: “Typical BCM5714S-Based LOM Design Block Diagram,” on page 41. • Figure 17: “Typical BCM5715C-Based LOM Design Block Diagram,” on page 43. • Figure 18: “Typical BCM5715S-Based LOM Design Block Diagram,” on page 45. • Figure 19: “Typical BCM5751-Based NIC Board Block Diagram,” on page 47. • Figure 20: “Functional Block Diagram,” on page 49. • Intelligent management device (IMD) to baseboard management controller (BMC) in “Universal Management Port (Applicable to BCM5714C/BCM5714S/ BCM5715C/BCM5715S Only)” on page 78. • Init for bit 8 in Table 171: “Power Management Control/Status Register (Offset 0x4C),” on page 347. • Table 185: “DMA Read/Write Control Register (Offset 0x6C),” on page 357. • Table 349: “Receive List Placement Statistics Enable Mask Register (Offset 0x2018),” on page 471. • 15-1 to 15-2 in Table 351: “Receive List Placement Statistics Increment Mask Register (Offset 0x201C),” on page 472. • Bits 31-30’s field in Table 423: “Read DMA Mode Register (Offset 0x4800),” on page 516. • Bits 31-30’s field in Table 426: “Write DMA Mode Register (Offset 0x4C00),” on page 519. • Bit 5’s description in Table 429: “RX RISC Mode Register Fields (Offset 0x5000),” on page 522. • Bit 5’s description in Table 433: “TX RISC Mode Register Fields (Offset 0x5400),” on page 526. • Bits 27-24’s field in Table 495: “Auxiliary SMBus Master Control Register (Offset 0x6C44),” on page 579. • Bit 0’s description in Table 498: “Auxiliary SMBus Slave Address/Control Register (Offset 0x6C50),” on page 581. • 0x6C60 to 0x6CE0 in: - “SMBus Address Resolution Protocol Registers (Offset 0x6CE0)” on page 584. - Table 501: “SMBus ARP Command Register (Offset 0x6CE0),” on page 584. • 0x6C64 to 0x6CE4 in: - “SMBus ARP Status Register (Offset0x6CE4)” on page 585. - Table 502: “SMBus ARP Status Register (Offset 0x6CE4),” on page 585. • 0x6C68 to 0x6CE8 in: - “UDID Register 0 (Offset 0x6CE8)” on page 586. - Table 503: “UDID Register 0 (Offset 0x6CE8),” on page 586. • 0x6C6C to 0x6CEC in: - “UDID Register 1 (Offset 0x6CEC)” on page 586. - Table 504: “UDID Register 1 (Offset 0x6CEC),” on page 586.

Revision	***Date***	***Description***
57XX-PG103-R (cont.)	08/18/05	**Updated (cont):** • 0x6C70 to 0x6CF0 in: - "UDID Register 2 (Offset 0x6CF0)" on page 586. - Table 505: "UDID Register 2 (Offset 0x6CF0)," on page 586. • 0x6C74 to 0x6CF4 in: - "UDID Register 3 (Offset 0x6CF4)" on page 586. - Table 506: "UDID Register 3 (Offset 0x6CF4)," on page 586. • In Table 573: "TLP Workaround Register (Offset 0x7C04, BCM5752 Only)," on page 624: - Bit 8 bit's field and description. - Bit 7's field and description. - Bit 3's field. - Bit 2's field. - Bit 1's field. - Bit 0's field and description. • Bit 18–24's fields and descriptions in Table 598: "Data Link Control Register (Offset 0x7D00)," on page 633. • Descriptions for bits 16–23 in Table 608: "Power Management Threshold Register (Offset 0x7D28)," on page 638. • Descriptions for bits 8, 9, and 11 in Table 630: "PHY Test Control Register (Offset 0x7E2C)," on page 647. **Added:** • Note below Table 1: "Family Features," on page 2. • "BCM5714C Dual-MAC Chip with Integrated Transceiver" on page 38. • "BCM5714S Dual-MAC chip with Integrated Fiber Transceiver" on page 40. • BCM5715 support to "RDI Timer Attention" on page 82. • Note at the end of "NVRAM Access Methods" on page 102. • Table 187: "PCI Clock Control Register (Offset 0x74)," on page 364. • Table 189: "PCI Clock Control Register Definition for BCM5705 Device," on page 367 through Table 191: "PCI Clock Control Register Definition for BCM5714, and BCM5715 Devices," on page 369. • Note to "MAC Function Register (0XB8H)" on page 378. • Table 206: "MAC Message Exchange Output Register (Offset 0xBC)," on page 378. • Table 228: "Virtual Channel Enhanced Capability Header (Offset 0x13c)," on page 393 through Table 250: "Firmware Power Budgeting Register 8 (Offset 0x18A)," on page 402. • BCM5705 and BCM5788 access for bit 12 in Table 256: "Ethernet MAC Status Register (Offset 0x404)," on page 413. • Table 262: "WOL Pattern Pointer Register (Offset 0x430, Rest of BCM57XXCT Family)," on page 419. • Table 264: "WOL Pattern Configuration Register (Offset 0x434, Rest of BCM57XXCT Family)," on page 420. • Table 291: "SerDes Receive Control Register (Offset 0x594, BCM5704S Only)," on page 435. • Table 298: "SerDes Transmit Control Register (0x5B4, BCM5714 and BCM5715 Only)," on page 440 through Table 311: "Statistics Registers," on page 444.

Revision	*Date*	*Description*
57XX-PG103-R (cont.)	08/18/05	**Added (cont):** • In Table 349: “Receive List Placement Statistics Enable Mask Register (Offset 0x2018),” on page 471: - Disable MACTQ double Ack issue fix to bit 18 - Disable ASF lockup fix to bit 1 • Table 360: “Receive Producer Ring NIC Address (Offset 0x244C),” on page 476. • These registers are applicable to BCM5700, BCM5701, BCM5702, BCM5703C, BCM5703S, BCM5704C, and BCM5704S devices only in “MBUF Cluster Free Registers” on page 486. • “DBU Registers” on page 487. • To Table 452: “Miscellaneous Local Control Register (Offset 0x6808),” on page 546: - GRC Alt Clock Enable to bit 31. - GRC Alt Clock Select to bit 30. • Table 492: “TPM Command Register (0x6c30, For BCM5714 and BCM5715 Only),” on page 577. • Table 493: “TPM Data Register (0x6C34),” on page 577. • Table 523: “NVM Status Register (0x7004H),” on page 597. • Table 539: “BIST Status Register (Offset 0x7404, BCM5721, BCM5751, and BCM5752 Only),” on page 606 through Table 543: “UART Register Map Summary,” on page 609. • Table 556: “UMP Registers (Applicable to BCM5714 and BCM5715 Only),” on page 614. • Table 572: “TLP Control Register (Offset 0x7C00, BCM5721, BCM5751, and BCM5752 Only),” on page 623. • “00h-0Fh 1000BASE-X Register Map Detailed Description” on page 667. • Table 412: “TX RISC MBUF Allocation Request Register (Offset 0x4424),” on page 511 through Table 414: “Hardware Diagnostic 1 Register (0x444Ch, 5714 only),” on page 513. • Table 416: “BM Hardware Diagnostic 2 Register (Offset 0x4450),” on page 513. • Table 417: “Hardware Diagnostic 2 Register (0x4450h, 5714 only),” on page 514. • “RX RISC Hardware Breakpoint Register (offset 0x5034)” on page 525. • Table 468: “Gig SerDes PRBS Control Register (0x6850, BCM5714 only),” on page 563 through Table 471: “GRC Message Exchange In Register (0x6874H, BCM5714 only),” on page 563. **Removed:** • In Table 185: “DMA Read/Write Control Register (Offset 0x6C),” on page 357: - Bit 31: Error Forwarding Error Enable - Bit 28: Byte Count Check • BCM5714 only support in Table 205: “MAC Function Register (0XB8H),” on page 378. • In Table 351: “Receive List Placement Statistics Increment Mask Register (Offset 0x201C),” on page 472: - Bit 18: Disable MACTQ Double Ack Issue Fix - Bit 1: Enable ASF Lockup Fix

Revision	*Date*	*Description*
57XX-PG102-R	09/30/04	**Updated:** • Table 216: “Ethernet MAC Status Register (Offset 0x404)” • Table 382: “RX RISC Registers” • Table 405: “Miscellaneous Configuration Register (Offset 0x6804)” • Table 406: “Miscellaneous Local Control Register (Offset 0x6808)” • Table 409: “RX-RISC Timer Reference Register (Offset 0x6814)” • Table 417: “Serial EEPROM Address Register (Offset 0x6838)” • Table 449: “Auxiliary SMBus Master Control Register (Offset 0x6C44)” through Table 454: “Auxiliary SMBus Slave Data Register (Offset 0x6C58)” • Table 476: “NVM Command Register (Offset 0x7000)” • Table 592: “PHY Identifier Registers (PHY_Addr = 0x1, Reg_Addresses 02h and 03h)”
57XX-PG101-R	02/04/04	**Updated:** • The BCM5721/BCM5751 registers to engineering register specification 2.6.
57XX-PG100-R	11/03/03	Initial release.

Broadcom Corporation
5300 California Avenue
Irvine, CA 92617

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Section 1: Introduction

INTRODUCTION

The BCM57XX NetXtreme® family is a family of triple-speed 10/100/1000BASE-T copper and gigabit 1000BASE-X fiber Ethernet LAN controller solutions for high-performance network applications. The devices are a highly-integrated solution combining the following functions:

- Triple-speed IEEE 802.3 compliant Media Access Controller (MAC) functions
- Peripheral Component Interconnect (PCI), PCI-X®, and PCI Express® (PCIe™) bus interfaces
- 10/100/1000BASE-T Ethernet transceiver
- 10/100/1000BASE-T Ethernet transceiver with SerDes
- On-chip packet buffer memory
- On-chip RISC processors for custom frame processing
- Universal Management Port (UMP) and SMBus interfaces for Alert Specification Function (ASF) and remote system management traffic
- NVRAM (SEEPROM and Flash) interface

Members of the NetXtreme BCM57XX family include:

- BCM5700 MAC
- BCM5701 MAC transceiver
- BCM5702 MAC transceiver
- BCM5703C MAC transceiver
- BCM5703S MAC transceiver SerDes
- BCM5704C dual-MAC transceiver
- BCM5704S dual-MAC transceiver SerDes
- BCM5705 MAC transceiver
- BCM5788 MAC transceiver
- BCM5721 MAC transceiver
- BCM5751 MAC transceiver
- BCM5714C dual-MAC transceiver
- BCM5714S dual-MAC transceiver SerDes
- BCM5715C dual-MAC transceiver
- BCM5715S dual-MAC transceiver SerDes
- BCM5752 MAC transceiver

Typical applications for the BCM57XX family include network interface cards (NICs) and LAN-on-motherboard (LOM).

Note: Refer to the *NetXtreme II™ Programmer's Guide* for BCM5706C, BCM5706S, and BCM5708 programming details.

FEATURE COMPARISON

Table 1 shows the features of the members of the BCM57XX family. See "Features" on page 11 for a more detailed description of the BCM57XX features.

Table 1: Family Features

Feature	***MAC Only***	***Single MAC Transceivers***								***Dual MAC Transceivers***			
	BCM5700	***BCM5701***	***BCM5702***	***BCM5703C***	***BCM5703S***	***BCM5705/ BCM5788***	***BCM5721***	***BCM5751***	***BCM5752***	***BCM5704C***	***BCM5704S***	***BCM5714C/ BCM5715C***	***BCM5714S/ BCM5715S***
Data Management													
VLAN tag support (IEEE 802.1q)	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Layer 2 priority encoding (802.1p)	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Link aggregation (IEEE 802.3ad)	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Full-duplex flow control (IEEE 802.3x)	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Programmable rules checker for advanced packet filtering and classification	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Frame/packet buffer memory[a]	96 KB	96 KB	96 KB	96 KB	96 KB	56 KB RX, 8 KB TX	64 KB RX, 8 KB TX	64 KB RX, 8 KB TX	64 KB RX, 8 KB TX	64 KB	64 KB	32 KB RX, 22 KB TX	32 KB RX, 22 KB TX
Jumbo frame support	9 KB	9 KB	9 KB	9 KB	9 KB	None	None	None	None	9 KB	9 KB	9 KB	9 KB
Calculation and verification of TCP, UDP, and IP checksums	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Multiple transmit and receive descriptor queues	Yes	Yes	Yes	Yes	Yes	No	No	No	No	Yes	Yes	No	No
Scatter/gather bus mastering architecture	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Auto-negotiation for 1000BASE-SX designs	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Statistics for SNMP MIB II, Ethernet like MIB, and Ethernet MIB (802.3z, Clause 30)	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Table 1: Family Features (Cont.)

Feature	MAC Only	Single MAC Transceivers								Dual MAC Transceivers			
	BCM5700	BCM5701	BCM5702	BCM5703C	BCM5703S	BCM5705/ BCM5788	BCM5721	BCM5751	BCM5752	BCM5704C	BCM5704S	BCM5714C/ BCM5715C	BCM5714S/ BCM5715S
IPMI support	No	No	No	Yes[j, k]	Yes [j, k]	No	Yes[k]	No	No	Yes[k]	Yes[k]	Yes[k]	Yes[k]
ASF support	No	No	No	Yes[l]	Yes[l]	No	Yes[m]	Yes[m]	Yes[m]	No	No	No	No
Trusted platform module support	No	No	No	No	No	No	No	Yes[n]	Yes	No	No	No	No
UMP interface	No	No	No	No	No	No	No	No	No	No	No	Yes	Yes
Bus Interfaces													
PCI v2.2 32-bit/64-bit, 33-MHz/66-MHz bus interface	Yes	Yes	32 bits only	Yes	Yes	32 bits only[o]	No	No	No	Yes	Yes	No	No
PCI-X v1.0 64-bit 100-MHz/133-MHz bus interface	Yes	Yes	No	Yes	Yes	No	No	No	No	Yes	Yes	No	No
CardBus interface	No	No	No	No	No	Yes	No	No	No	No	No	No	No
PCIe v1.0a x1 bus interface	No	No	No	No	No	No	Yes	Yes	Yes	No	No	No	No
PCIe v1.0a x4 bus interface	No	No	No	No	No	No	No	No	No	No	No	Yes	Yes
PCIe v1.0a x2 bus interface	No	No	No	No	No	No	No	No	No	No	No	Yes	Yes
LAN Interfaces													
10/100/1000BASE-T full-duplex/half-duplex MAC	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Integrated 10/100/1000BASE-T transceiver	No[c]	Yes[e]	Yes[e]	Yes[e]	Yes[e]	Yes[e]	Yes[e]	Yes[e]	Yes[e]	Yes[e]	Yes[e]	Yes[e]	Yes[e]
Internal MII/GMII	Yes	Yes	Yes[f]	Yes[f]	No	Yes[f]	Yes[f]	Yes[f]	Yes[f]	Yes[f]	No	Yes	No
Internal GMII/MII	Yes	Yes	Yes[g]	Yes[g]	Yes[g]	Yes[g]	Yes[g]	Yes[g]	Yes[g]	Yes[g]	Yes[g]	Yes[g]	Yes
TBI (SerDes)	Yes	Yes	No	No	Yes	No	No	No	No	No	Yes	No	Yes
Self-Test													
Test modes (BIST, SCAN, and so on)	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Table 1: Family Features (Cont.)

Feature	***MAC Only***	***Single MAC Transceivers***								***Dual MAC Transceivers***			
	BCM5700	***BCM5701***	***BCM5702***	***BCM5703C***	***BCM5703S***	***BCM5705/ BCM5788***	***BCM5721***	***BCM5751***	***BCM5752***	***BCM5704C***	***BCM5704S***	***BCM5714C/ BCM5715C***	***BCM5714S/ BCM5715S***
Factory-level JTAG support	Yes	Yes	Yes[h]	Yes[h]	Yes[h]	Yes[h]	Yes[h]	Yes[h]	Yes[h]	Yes[h]	Yes[h]	Yes[h]	Yes[h]
Firmware													
SMBus 2.0 interface	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Interface to Flash memory	No	No	Yes[i]	Yes[i]	Yes[i]	Yes[i]	Yes[i]	Yes[i]	Yes[i]	Yes[i]	Yes[i]	Yes[i]	Yes[i]
Interface to serial EEPROM	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Firmware TCP segmentation	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Technology													
High-performance, low overhead, SW/HW interface	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
High speed on-chip RISC processors[b]	Dual	Dual	Dual	Dual	Dual	Single	Single	Single	Single	Dual	Dual	Single	Single
WOL support meeting the ACPI requirements	Yes[d]	Yes[d]	Yes[d]	Yes[d]	Yes[d]	Yes[d]	Yes[d]	Yes[d]	Yes[d]	Yes[d]	Yes[d]	Yes[d]	Yes[d]
Maximum external SSRAM	16 MB	None	None	None	None	None	None	None	None	None	None	None	None
Process voltage	3.3V/1.8 V	3.3 V/1.8 V	1.2V	1.2V	1.2V	1.2V	1.2V	1.2V	1.2V	1.2V	1.2V	1.2V	1.2V
CMOS linewidth	0.18 µm	0.18 µm	0.13 µm	0.13 µm	0.13 µm	0.13 µm	0.13 µm	0.13 µm	0.13 µm	0.13 µm	0.13 µm	0.13 µm	0.13 µm
5V tolerant PCI I/Os	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

[a] Memory amount given is per PCI function.

[b] Indicates the number of processors per port.
Example: Each port of the BCM5704C has two RISC processors, for a total of four RISC processors.

[c] Separate PHY required (BCM5401 PHY).

[d] Conformance to PCI Power Management specification v1.1, supporting:
- Magic Packet wake-up
- Wake-up on interesting packet

e. Automatic MDI crossover function
- Automatic detection and correction of pair swaps, pair skew, and pair polarity;
- Auto-negotiation with Next Page capability

f. These devices have internal PHYs, with the MII/GMII internal between the MACs and PHYs. There is no external MII/GMII between the MACs and PHYs.

g. The GMII/MII management interface is managed internally using indirect register access via the MAC (see "Transceiver Registers" on page 603). There are no MDIO, MDC, or MDINT ports.

h. Since these chips use 0.13 µm process silicon, the JTAG is only 3.3V tolerant.

i. Up to 16 MB of Flash memory supported.

j. Requires revision B0 silicon or later.

k. Supports IPMI 1.5 specification through optional firmware.

l. Supports ASF 1.03 specification through optional firmware.

m. Supports ASF 2.0 specification through optional firmware.

n. Supported on BCM5751T and BCM5751TM parts.

o. The BCM5705 supports dual addressing as per PCI specification, whereby a type of 64-bit addressing is possible. The BCM5788 does not support dual addressing.

Note: The BCM5752 does not support VPD writes.

REVISION LEVELS

See Table 2 for the revision levels of the BCM57XX family. Host software can use the PCI Revision ID and Chip ID information in the PCI configuration registers to determine the revision level of the BCM57XX chip on the board to load the appropriate workaround described in the errata sheets.

The Broadcom PCI vendor ID is 0x14E4. Table 2 shows the default values of PCI device IDs. These values may be modified by firmware in accordance with the manufacturing information supplied in NVRAM (see Section 4: "NVRAM Configuration" on page 88 for more details).

Table 2: Family Revision Levels

Family Member	*Device ID*[a]	*Revision Level*	*PCI Revision ID*[b]	*Chip ID*[c]	*PHY Core*	*Errata Sheet*[d]
BCM5700 MAC	0x1644	B0	0x10	0x7100xxxx	N/A	5700-ES3XX-R
	0x1644	B1	0x11	0x7101xxxx	N/A	5700-ES3XX-R
	0x1644	B2	0x12	0x7102xxxx	N/A	5700-ES1XX-R 5700-ES8XX-R
	0x1644	B3	0x13	0x7103xxxx	N/A	5700-ES4XX-R
	0x1644	C0	0x20	0x7200xxxx	N/A	5700-ES5XX-R
	0x1644	C1	0x21	0x7201xxxx	N/A	5700-ES6XX-R
	0x1644	C2	0x22	0x7202xxxx	N/A	5700-ES7XX-R
BCM5701 MAC transceiver	0x1645	A0	0x08	0x0000xxxx	BCM5402 C0	5701-ES1XX-R
	0x1645	A2	0x12	0x0002xxxx	BCM5402 C0	5701-ES2XX-R
	0x1645	A3	0x15	0x0003xxxx	BCM5402 C0	5701-ES3XX-R
	0x1645	B5	0x25	0x0105xxxx	BCM5402 C2	–
BCM5702 MAC transceiver	0x16A6	A0	0x00	0x1000xxxx	BCM5421 A1	5702-ES1XX-R
	0x16A6	A1	0x01	0x1001xxxx	BCM5421 A2	–
	0x16A6	A2	0x02	0x1002xxxx	BCM5421 A2	5702-ES3XX-R

Table 2: Family Revision Levels (Cont.)

Family Member	Device ID[a]	Revision Level	PCI Revision ID[b]	Chip ID[c]	PHY Core	Errata Sheet[d]
BCM5703C MAC transceiver	0x16A7	A0	0x00	0x1000xxxx	BCM5421 A1	5703C-ES1XX-R
	0x16A7	A1	0x01	0x1001xxxx	BCM5421 A2	–
	0x16A7	A2	0x02	0x1002xxxx	BCM5421 A2	5703C-ES3XX-R
	0x16C7	B0	0x10	0x1100xxxx	BCM5421 A3	5703C-ES4*XX*-R
	0x16A7	B1	0x02	0x1002xxxx	BCM5421 A3	–
BCM5703S MAC transceiver SerDes	0x16A7	A0	0x00	0x1000xxxx	N/A	5703S-ES1XX-R
	0x16A7	A1	0x01	0x1001xxxx	N/A	–
	0x16A7	A2	0x02	0x1002xxxx	N/A	5703S-ES3XX-R
	0x16C7	B0	0x10	0x1100xxxx	N/A	5703S-ES4*XX*-R
	0x16A7	B1	0x02	0x1002xxxx	N/A	–
BCM5704C dual MAC transceiver	0x1648	A0	0x00	0x2000xxxx	BCM5421 A1	5704C-ES1XX-R
	0x1648	A1	0x01	0x2001xxxx	BCM5421 A1	5704C-ES2XX-R
	0x1648	A2	0x02	0x2002xxxx	BCM5421 A1	5407C-ES3XX-R
	0x1648	A3	0x03	0x2003xxxx	BCM5421 A1	5704C-ES4XX-R
	0x1648	B0	0x10	0x2100xxxx	BCM5421 A1	5704C-ES5XX-R
BCM5704S dual MAC transceiver SerDes	0x16A8[e]	A0	0x00	0x2000xxxx	N/A	–
	0x16A8[e]	A1	0x01	0x2001xxxx	N/A	5704S-ES1XX-R
	0x16A8[e]	A2	0x02	0x2002xxxx	N/A	5704S-ES2XX-R
	0x16A8[e]	A3	0x03	0x2003xxxx	N/A	5704S-ES2XX-R
	0x16A8[e]	B0	0x10	0x2100xxxx	N/A	5704S-ES3XX-R
BCM5705 MAC transceiver	0x1653	A0	0x00	0x3000xxxx	BCM5464 A0	5705-ES1XX-R
	0x1653	A1	0x01	0x3001xxxx	BCM5464 A0	5705-ES2XX-R
	0x1653	A2	0x02	0x3002xxxx	BCM5464 A0	–
	0x1653	A3	0x03	0x3003xxxx	BCM5464 A0	5705-ES3XX-R
	0x1653	A5	0x05	0x3005xxx	BCM5464 A0	5705-ES3XX-R
BCM5705M MAC transceiver	0x165d	A1	0x01	0x3001xxxx	BCM5464 A0	5705M-ES2XX-R
	0x165e	A2	0x02	0x3002xxxx	BCM5464 A0	–
	0x165e	A3	0x03	0x3003xxxx	BCM5464 A0	5705M-ES3XX-R
	0x165e	A5	0x05	0x3005xxxx	BCM5464 A0	5705M-ES4XX-R
BCM5788 MAC transceiver	0x169C	A5	0x05	0x3005xxxx	BCM5464 A0	5788-ES2XX-R
BCM5721 MAC transceiver	0x1659	A0	0x00	0x4000xxxx	BCM5464 egphy_core_hd-A103	5721-ES1XX-R
	0x1659	A1	0x01	0x4001xxxx	BCM5464 egphy_core_hd-A103	5721-ES2XX-R
BCM5751 MAC transceiver	0x1677	A0	0x00	0x4000xxxx	BCM5464 egphy_core_hd-A103	5751-ES1XX-R
	0x1677	A1	0x01	0x4001xxxx	BCM5464 egphy_core_hd-A103	5751-ES2XX-R
BCM5751M MAC transceiver	0x167d	A0	0x00	0x4000xxxx	BCM5464 egphy_core_hd-A103	5751M-ES1XX-R
	0x167d	A1	0x01	0x4001xxxx	BCM5464 egphy_core_hd-A103	5751M-ES2XX-R

Table 2: Family Revision Levels (Cont.)

Family Member	Device ID[a]	Revision Level	PCI Revision ID[b]	Chip ID[c]	PHY Core	Errata Sheet[d]
BCM5752 MAC transceiver	0x1600	A0	0x00	0x5000xxxx	BCM5464 gphy_core_hd-A104	
	0x1600	A1	0x01	0x6001xxxx	BCM5464 gphy_core_hd-A104	5752-ES2XX-R
BCM5752M MAC transceiver	0x1601	A0	0x00	0x5000xxxx	BCM5464 gphy_core_hd-A104	
	0x1601	A1	0x01	0x6001xxxx	BCM5464 gphy_core_hd-A104	5752M-ES2XX-R
BCM5714C MAC transceiver	0x1668	A0	0x00	0x5000xxxx	BCM5464 gphy_core_hd-A104	5714C-ES02-R
	0x1668	A1	0x01	0x9001xxxx	BCM5464 gphy_core_hd-A104	5714C-ES2XX-R
BCM5714S MAC transceiver	0x1669	A0	0x00	0x5000xxxx	BCM5464 gphy_core_hd-A104	5714S-ES02-R
	0x1669	A1	0x01	0x9001xxxx	BCM5464 gphy_core_hd-A104	5714S-ES2XX-R
BCM5715C MAC transceiver	0x1678	A0	0x00	0x5000xxxx	BCM5464 gphy_core_hd-A104	5715C-ES02-R
	0x1678	A1	0x01	0x9001xxxx	BCM5464 gphy_core_hd-A104	5715C-ES2XX-R
BCM5715S MAC transceiver	0x1679	A0	0x00	0x5000xxxx	BCM5464 gphy_core_hd-A104	5715S-ES02-R
	0x1679	A1	0x01	0x9001xxxx	BCM5464 gphy_core_hd-A104	5715S-ES2XX-R

[a] See "Device ID Register (Offset 0x02)" on page 301.
[b] See "Revision ID Register (Offset 0x08)" on page 304. Broadcom firmware programs this value after device reset. The hardware default is 0x00.
[c] See "Miscellaneous Host Control Register (Offset 0x68)" on page 325. The lower 16 bits are don't cares.
[d] See the appropriate errata documentation for the errata information and resolutions.
[e] The device ID is 0x16A8 if boot code is v3.04 or later; otherwise it is 0x1648.

PROGRAMMING THE BCM57XX FAMILY

The reference documents for BCM57XX software development include this manual and the errata documentation (see "Revision Levels" on page 5) that provide the necessary information for writing a host-based device driver.

The programming model for the BCM57XX MAC does not have interdependency upon OS or processor instruction sets. Programmers using Motorola® 68000, Intel® x86, or DEC Alpha host instruction sets will be able to leverage this document to aid and assist in device driver development. Additionally, concepts provided in this document are applicable to device drivers native to any operating system (i.e., DOS, UNIX®, Microsoft®, or Novell®).

About This Manual

The Host Programmer Reference manual provides information on how to write device drivers for the BCM57XX NetXtreme family. This manual focuses on the registers, control blocks, and software interfaces necessary for host software programming. This document is intended to complement the data sheet for the appropriate member of the BCM57XX family. The errata documentation (see "Revision Levels" on page 5) complements this manual.

Related Documentation

Data sheets, application notes, and errata documentation for:

- BCM5700
- BCM5701
- BCM5702
- BCM5703C
- BCM5703S
- BCM5704C
- BCM5704S
- BCM5705
- BCM5788
- BCM5721
- BCM5751
- BCM5714C
- BCM5714S
- BCM5715C
- BCM5715S
- BCM5752

Notational Conventions

Registers and Bits

Register and bit names are concatenated with underscores:

- Mac_Mode (register)
- Enable_TDE (bit)

Periods separate a register and bit pair:

- Register.Bit
- Mac_Mode.Enable_TDE

Documentation generally avoids referencing bits by offset; the register definition section provides register and bit offsets.

Functional Overview

Functional descriptions provide high level overviews of the BCM57XX architectural blocks. The black box inputs/outputs from block diagrams aid software developers with programming the device.

Operational Characteristics

This section describes how software programs or interfaces with a hardware block.

Pseudocode

Table 3 defines the C style pseudocode that is used throughout this document. Pseudocode is an informal syntax intended to explain complex algorithms. The methods used to program the BCM57XX family and companion silicon may be defined using pseudocode. Table 3 is not meant to restrict algorithmic representations, but help define a structure used throughout the document. This pseudocode is not a formal language and does not have a formal grammar. When appropriate, pseudocode may deviate from these notations.

Table 3: Pseudocode

Definition	*Notation*	*Notes*
Register and bit field	register.bit	Bold
Variable	variable	Italics
Pointer	variablePtr	Italics + Ptr
Constant	CONSTANT_DEFINITION	Capitalization
<Conditional>	If <Expression> Then <Block> Else <Block>	

<Block>	Begin or Cstyle {
	<Pseudo Code>
	End or Cstyle}
		Any valid logic including <Expression>, <Repetition>, <While>, <Conditionals>, <Assignment>, etc.

<While>	While <expression> <Block>	Other variants of while valid (i.e., Do while, and so on).
<Repetition>	For < iterations> do < Block>	
<Expression>	<Expression1> <Operator> <Expression2>	Apply operator to two subexpressions E1 and E2. The result will be interpreted as a boolean value
<Assignment>	variable/register = CONSTANT variable/register = variable variable/register = <Expression> variable/register = variable/register <Operator>variable/register	assign constant assign to variable assign to result of expression (Boolean) assign to result of operation on variables/ registers

Table 3: Pseudocode (Cont.)

Definition	***Notation***	***Notes***
<Operator>	&	Bitwise And
	|	Bitwise Or
	<<	Bitwise Shift Left
	>>	Bitwise Shift Right
	==	Boolean equality
	!=	Boolean inequality
	<	Less than
	<=	Less than equal
	>	Greater than
	>=	Greater than equal
	And	Boolean And
	Or	Boolean Or
Procedure/subroutine	Procedure <name>	
Call procedure/subroutine	Call <name>	
Operational comment	// Comments, Comments, Comments	Not part of the logic flow
Deference	*(variable/register/CONSTANT _DEFN)	Contents of an address/location

Section 2: Features

Typical applications for the BCM57XX NetXtreme family include NICs and a LOM.

BCM5700 MAC

The BCM5700 is the first generation Gigabit Ethernet Controller device with a 10/100/1000 Mbps Ethernet MAC that supports full-duplex/half-duplex capability at all speeds. The MAC interfaces with the external transceiver via the Gigabit Media Independent Interface (GMII) in 1000BASE-T mode, the Media Independent Interface (MII) in 10/100BASE-T mode, and 10-bit interface (TBI) in SerDes mode. The device provides both PCI v2.2 and PCI-X v1.0 bus interfaces. The BCM5700 MAC provides large on-chip buffer memory for stand-alone operation and optionally supports up to 16 MB of external memory. Dual on-chip high-performance RISC processors enable custom frame processing features including TCP segmentation, IP fragmentation, and IP reassembly. The device is fabricated in a low-voltage 1.8V CMOS process providing a low-power system solution. Following are the important features of BCM5700 MAC device.

- PCI-v2.2 32-bit/64-bit, 33-MHz/66-MHz bus interface
- PCI-X v1.0 64-bit, 133-MHz bus interface
- Dual high-speed on-chip RISC processors
- 10/100/1000BASE-T full-duplex and half-duplex MAC
- MII/GMII and TBI MAC-PHY interfaces
- IEEE 802.1Q VLAN tag support
- IEEE 802.1p Layer2 priority encoding
- WOL support meeting ACPI requirements
- IEEE 802.3x flow control
- Integrated 96-KB packet buffer memory
- Programmable receive rule checker
- 9KB jumbo frame support
- Statistics for SNMP MIB II, Ethernet-like MIB, and Ethernet MIB (802.3z Clause 30)
- 16-MB external synchronous static RAM (SSRAM) support
- MDIO management interface
- Serial EEPROM interface
- JTAG support
- 3.3V/1.8V CMOS with 5V tolerant PCI I/Os
- 388 PBGA package

A complete NIC or LOM solution for Gigabit Ethernet is achieved by combining the BCM5700 MAC with the BCM5401 single-chip Gigabit Ethernet transceiver (see Figure 1).

TYPICAL APPLICATION (COPPER-BASED LAYOUT)

The following figure shows a typical BCM5700-based NIC board layout.

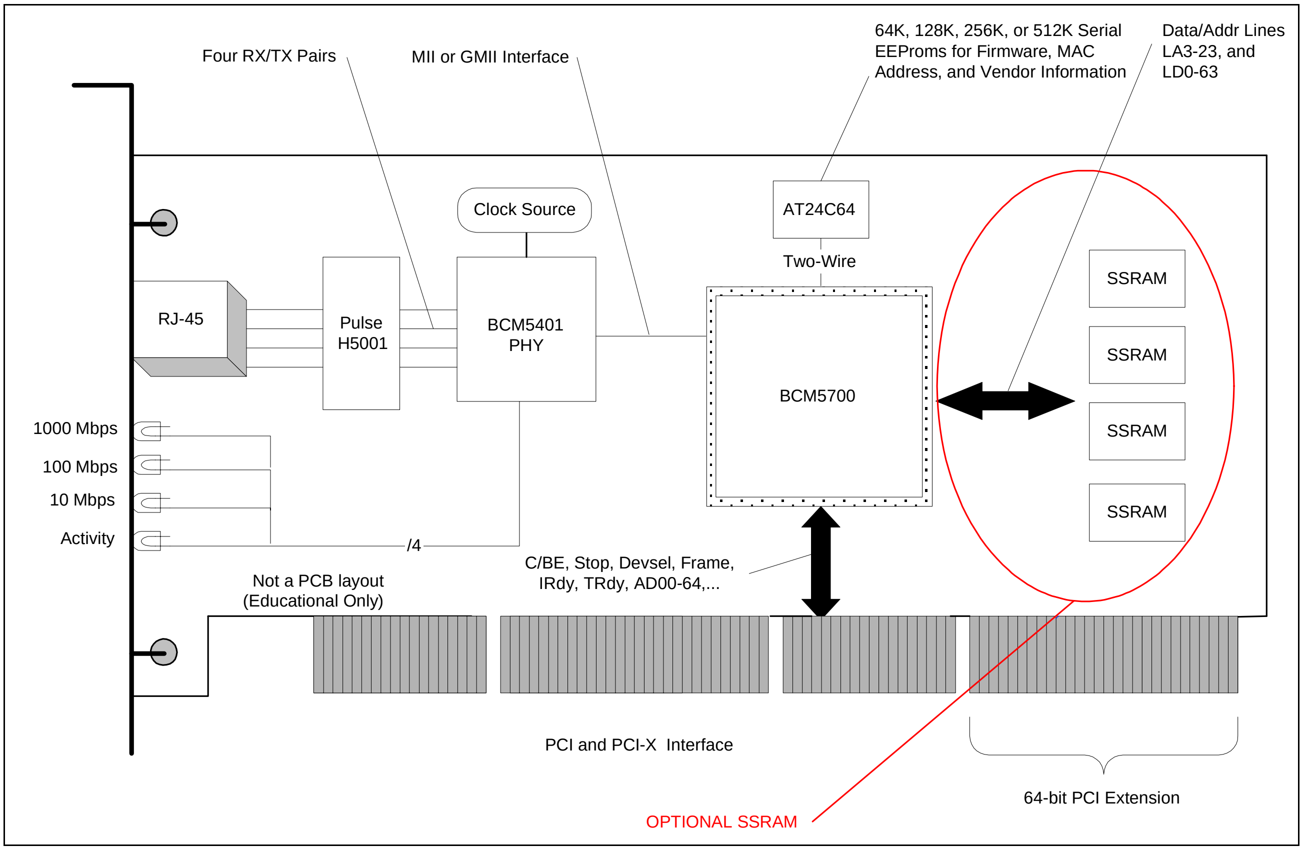


Figure 1: Typical BCM5700-Based NIC Board Layout

The following table lists the part component breakdown.

Table 4: BCM5700 NIC Part Component Breakdown

Part Component	*Description*
RJ45	The physical connector for category 5 twisted-pair cabling
Magnetics	The pulse H5001 isolates the physical layer from voltage events such as sags, swells, and transients. The magnetics module also compensates for impedance mismatches between the cabling and physical layer.
PHY	The BCM5401 physical layer (PHY) component converts analog waveforms to digital signals. A DSP processes analog signals and compensates for return loss, cable loss, far-end cross talk, and near-end cross-talk. The digital data is passed to the MAC via the GMII or MII connection.
Clock source	A crystal oscillator generates a 25-MHz clocking signal. This clock is an input to the physical layer component. A PLL in the PHY drives a 125-MHz GMII clock to the MAC.
Voltage regulators	The BCM95700 reference design provides three voltage sources for the MAC and PHY components. The BCM5700 MAC uses a 3.3V power rail for PCI I/O drivers and a 1.8V power rail for core logic. The PHY uses all 3.3V, 2.5V, and 1.8V power rails. The voltage regulators will step voltage down from 5V or 3.3V slot sources to the three respective levels.

Table 4: BCM5700 NIC Part Component Breakdown (Cont.)

Part Component	*Description*
MAC	The BCM5700 MAC.
GMII/MII	• GMII is an 8-bit wide interface that is clocked at 125 MHz. The resultant bandwidth is 1 Gbps. • MII is 4 bits wide and clocks at 25 MHz.
SSRAM	External SSRAM is optional. Applications may attach 16 MB of external SSRAM to the BCM5700 MAC. 8 MB can be made available to packet RX/TX pools and 8 MB for RISC addressable memory. The SSRAM banks must be 133-MHz pipelined SSRAM with a 5-ns maximum access time.
PCI/PCI-X	The PCI specification defines a protocol for bus master controller and target data movement. The MAC is a bus master controller and may move data without CPU intervention. The BCM5700 MAC supports both the PCI v2.2 and the PCI-X v1.0 specifications.
EEPROM	The non-volatile storage for firmware boot code, MAC address, VPD, and PXE code. The EEPROM is read after the BCM5700 MAC is reset.

TYPICAL APPLICATION (FIBER-OPTIC LAYOUT)

The following figure shows a typical BCM5700-based SerDes board layout.

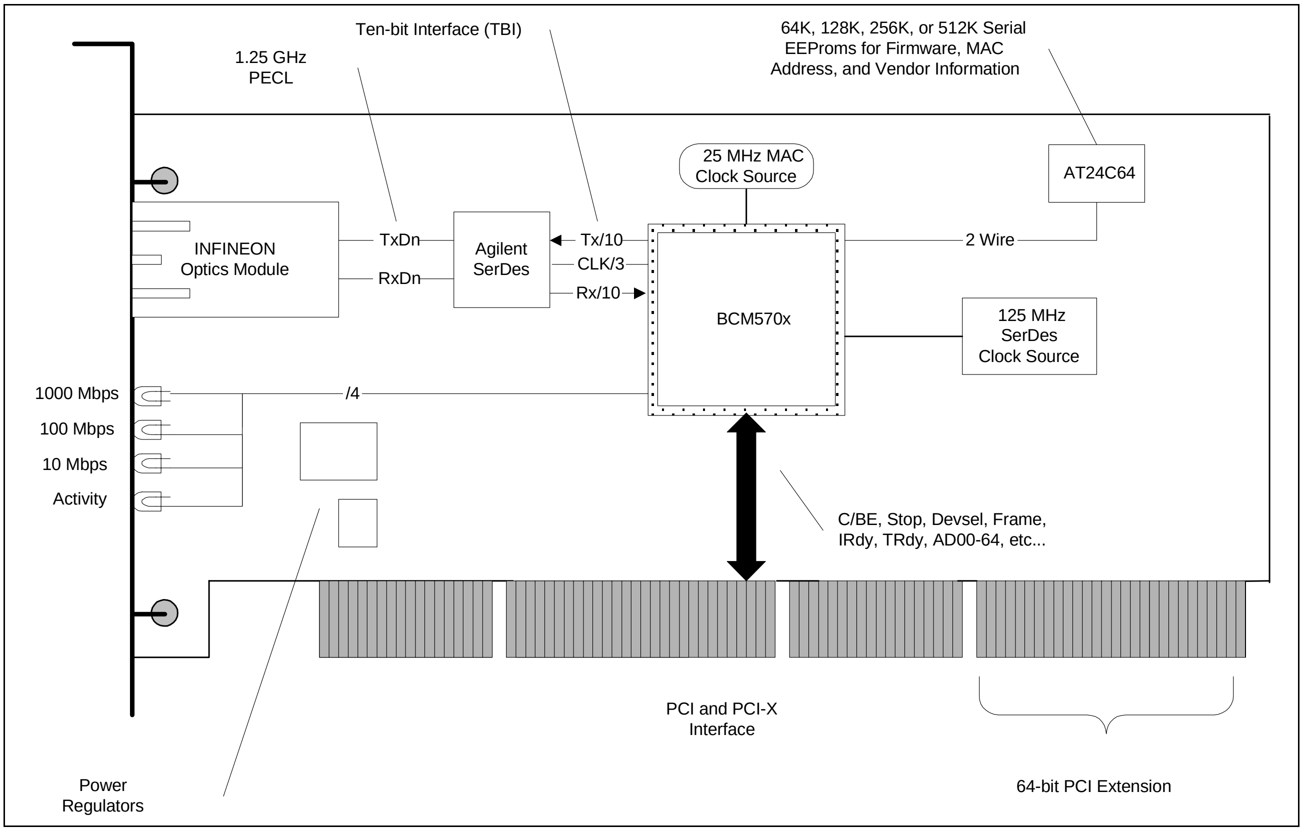


Figure 2: Typical BCM5700-Based Fiber-Optic Board Block Diagram

The following table lists the part component breakdown.

Table 5: BCM5700 Fiber-Optic Part Component Breakdown

Part Component	Description
Optics module	Multimode or single-mode fiber optic physical layer.
SerDes	The serializer-deserializer accepts TBI input and outputs a 1.25-GHz PECL output.
Clock source	Two clock sources are used in the fiber application: a 125-MHz clock for the TBI interface and a 25-MHz clock for core MAC functionality (i.e., RX, TX, PCI).
Voltage regulators	Three external voltage regulators for 3.3V, 2.5V, and 1.8V (the Agilent SerDes chip uses 3.3V power source).
MAC	The BCM5700 MAC.
PCI/PCI-X	The PCI specification defines a protocol for bus master controller and target data movement. The MAC is a bus master controller and may move data without CPU intervention. The BCM5700 MAC supports both the PCI v2.2 and the PCI-X v1.0 specifications.
EEPROM	The non-volatile storage for firmware boot code, MAC address, VPD, and PXE code. The EEPROM is read after the BCM5700 MAC is reset.

PROGRAMMING ASPECTS

See Table 2: "Family Revision Levels," on page 5 for the revision levels of the BCM57XX family. Host software can use the PCI Revision ID and Chip ID information in the PCI configuration registers to determine the revision level of the BCM57XX chip on the board to load the appropriate workarounds described in the errata sheets.

Choice of host access mode determines mailbox priority (see "Configuration Space" on page 178 in Section 9: "PCI"):

- Host standard or flat mode uses the high-priority mailboxes (see "High-Priority Mailboxes" on page 370).
- Indirect mode uses the low-priority mailboxes (see "Low-Priority Mailboxes" on page 490).

Mini rings require the presence of external SSRAM. External SSRAM is option for the BCM5700 MAC, but external SSRAM must be present for the BCM5700 MAC to support mini rings.

See "Device Control" on page 146 for the procedure to initialize this device.

BCM5701 MAC WITH INTEGRATED TRANSCEIVER

The BCM5701 is the second generation Gigabit Ethernet Controller that integrates the triple speed MAC and 10/100/1000BASE-T Ethernet transceiver into one device. The integrated Ethernet PHY transceiver performs all the physical layer functions for 1000BASE-T, 100BASE-T, and 10BASE-T Ethernet on standard Category 5 UTP. The BCM5701 MAC also supports a TBI for 1000BASE-SX connections. Based on Broadcom's proven DSP technology, the integrated Ethernet PHY transceiver device is a highly integrated solution combining digital adaptive equalizers, ADCs, PLLs, line drivers, echo cancellers, crosstalk cancellers, and all other required support circuitry.

The BCM5701 MAC provides both PCI v2.2 and PCI-X v1.0 bus interfaces, as well as a large on-chip buffer memory. Dual on-chip high-performance processors enable custom frame processing features such as TCP segmentation. A full-featured MAC provides full-duplex/half-duplex capability at all speeds. The device is fabricated in a low-voltage 1.8V CMOS process providing a low-power system solution.

- PCI-v2.2 32-bit/64-bit, 33-MHz/66-MHz bus interface
- PCI-X v1.0 64-bit, 133MHz bus interface
- Dual high-speed on-chip RISC processors
- 10/100/1000BASE-T full-duplex and half-duplex MAC
- 10/100/1000BASE-T Ethernet PHY transceiver
- TBI MAC-PHY interface for 1000-BASE SX Connections
- IEEE 802.1Q VLAN tag support
- IEEE 802.1p Layer2 priority encoding
- WOL support meeting ACPI requirements
- IEEE 802.3x flow control
- Integrated 96-KB packet buffer memory
- Programmable receive rule checker
- 9KB jumbo frame support
- Statistics for SNMP MIB II, Ethernet-like MIB, and Ethernet MIB (802.3z Clause 30)
- Serial EEPROM interface
- JTAG support
- 1.8V CMOS with 5V tolerant PCI I/Os
- 300 BGA and 304 PBGA packages

Note: The BCM5701 and later MAC transceivers include an integrated MAC and PHY. Host driver software can program the integrated PHY just as if it were an external discrete chip. Host software can use the traditional bit-bang method or auto-access methods to access PHY MDI register space (see "Access Methods" on page 248).

To create the Broadcom one-chip solution, different PHY cores have been integrated with different BCM57XX parts as listed in Table 2 on page 5. The BCM57XX MAC and PHY cores will be enhanced and changed independently of their respective baseline parts.

The BCM5701 MAC transceiver provides a complete NIC or LOM solution for Gigabit Ethernet (see Figure 3).

TYPICAL APPLICATION

The following figure shows a typical BCM5701-based NIC board layout.

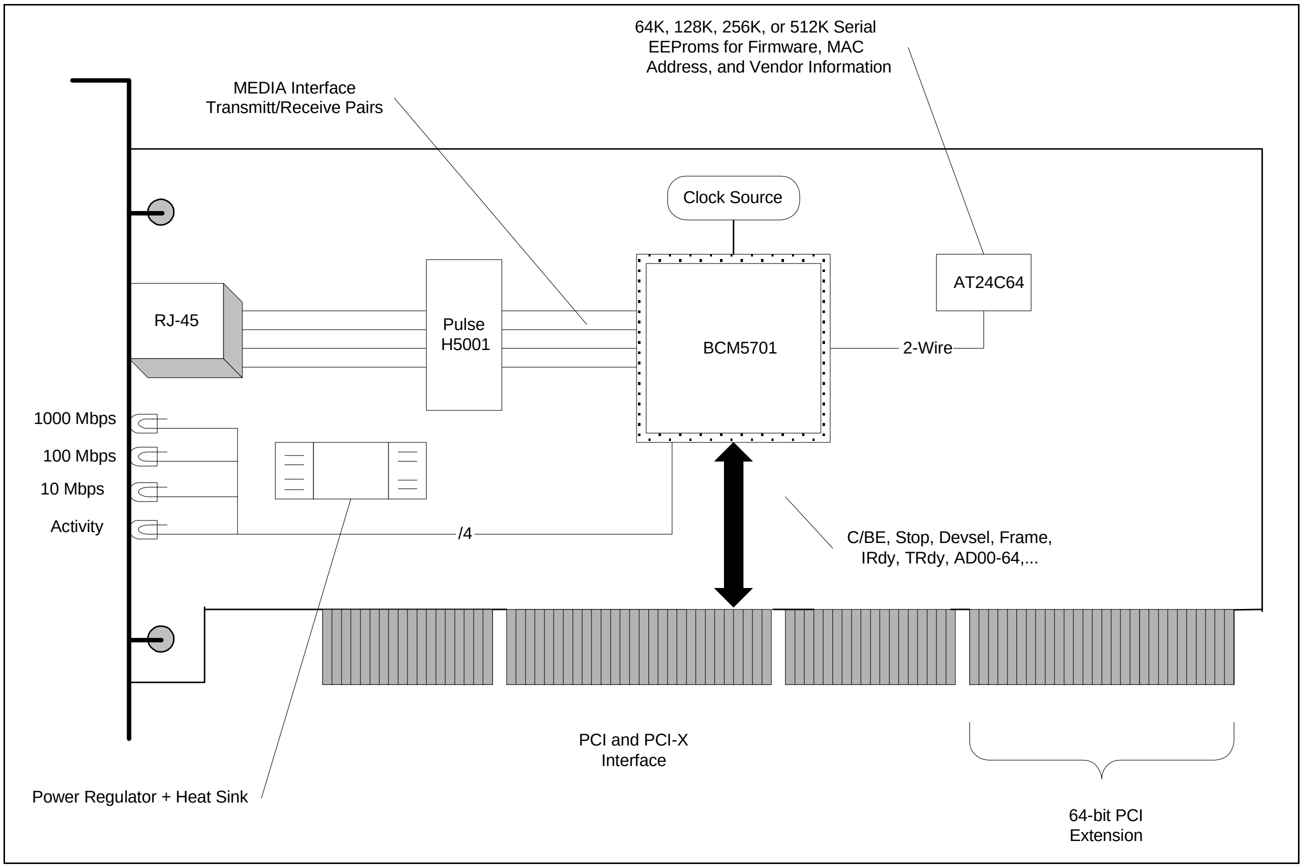


Figure 3: Typical BCM5701-Based NIC Board Block Diagram

The following table lists the part component breakdown.

Table 6: BCM5701 NIC Part Component Breakdown

Part Component	***Description***
RJ45	The physical connector for category 5 twisted-pair cabling.
Magnetics	The pulse H5001 isolates the physical layer from voltage events such as sags, swells, and transients. The magnetics module also compensates for impedance mismatches between the cabling and physical layer.
Clock source	A crystal oscillator generates a 25-MHz clocking signal.
Voltage regulators	The BCM95701 reference design provides three voltage sources for the BCM5701 MAC-transceiver: 3.3V, 2.5V, and 1.8V. The voltage regulators will step voltage down from 5V or 3.3V slot sources to the three respective levels.
MAC	The BCM5701 MAC.
PCI/PCI-X	The PCI specification defines a protocol for bus master controller and target data movement. The MAC is a bus master controller and may move data without CPU intervention. The BCM5701 MAC supports both the PCI v2.2 and the PCI-X v1.0 specifications.
EEPROM	The non-volatile storage for firmware boot code, MAC address, VPD, and PXE code. The EEPROM is read after the MAC is reset.

PROGRAMMING ASPECTS

See Table 2: "Family Revision Levels," on page 5 for the revision levels of the BCM57XX family. Host software can use the PCI Revision ID and Chip ID information in the PCI configuration registers to determine the revision level of the BCM57XX chip on the board to load the appropriate workarounds described in the errata sheets.

The BCM5701 MAC-Controller is an integrated MAC and PHY. The BCM5402 PHY was integrated together with the triple speed BCM5700 MAC core to create the Broadcom one-chip solution—the BCM5701 MAC transceiver. The BCM5701 MAC core and BCM5402 PHY cores will be enhanced/changed independently of their respective baseline parts (BCM5402 PHY and BCM5700 MAC). Host driver software can program the integrated PHY just as if it were an external discrete/chip. Host software can use the traditional bit-bang method or auto-access methods to access PHY MDI register space (see "Access Methods" on page 248).

Choice of host access mode determines mailbox priority (see "Configuration Space" on page 178 in Section 9: "PCI"):

- Host standard or flat mode uses the high-priority mailboxes (see "High-Priority Mailboxes" on page 370).
- Indirect mode uses the low-priority mailboxes (see "Low-Priority Mailboxes" on page 490).

Mini rings require the presence of external SSRAM. Since external SSRAM is not supported by the BCM57XX family except for the BCM5700 MAC, mini rings are not supported on this version of the chip.

See "Device Control" on page 146 for the procedure to initialize this device.

BCM5702 MAC WITH INTEGRATED TRANSCEIVER

The BCM5702 is a third generation device that combines a triple speed MAC with a triple speed Ethernet transceiver, a 32-bit PCI, and 96KB Frame Buffer Memory. Following are the important features of BCM5702 device.

- PCI-V2.2 32-bits, 33-MHz/66-MHz bus interface
- Dual high-speed on-chip RISC processors
- 10/100/1000BASE-T full-duplex and half-duplex MAC
- 10/100/1000BASE-T Ethernet PHY transceiver
- IEEE 802.1Q VLAN tag support
- IEEE 802.1p Layer2 priority encoding
- WOL support meeting ACPI requirements
- IEEE 802.3x flow control
- Integrated 96-KB packet buffer memory
- Programmable receive rule checker
- 9KB jumbo frame support
- Statistics for SNMP MIB II, Ethernet-like MIB, and Ethernet MIB (802.3z Clause 30)
- Serial EEPROM and serial Flash support
- JTAG support
- 1.2V CMOS with 5V tolerant PCI I/Os
- Available in a 196 FPBGA package

TYPICAL APPLICATION

The following figure shows a typical BCM5702-based NIC board layout.

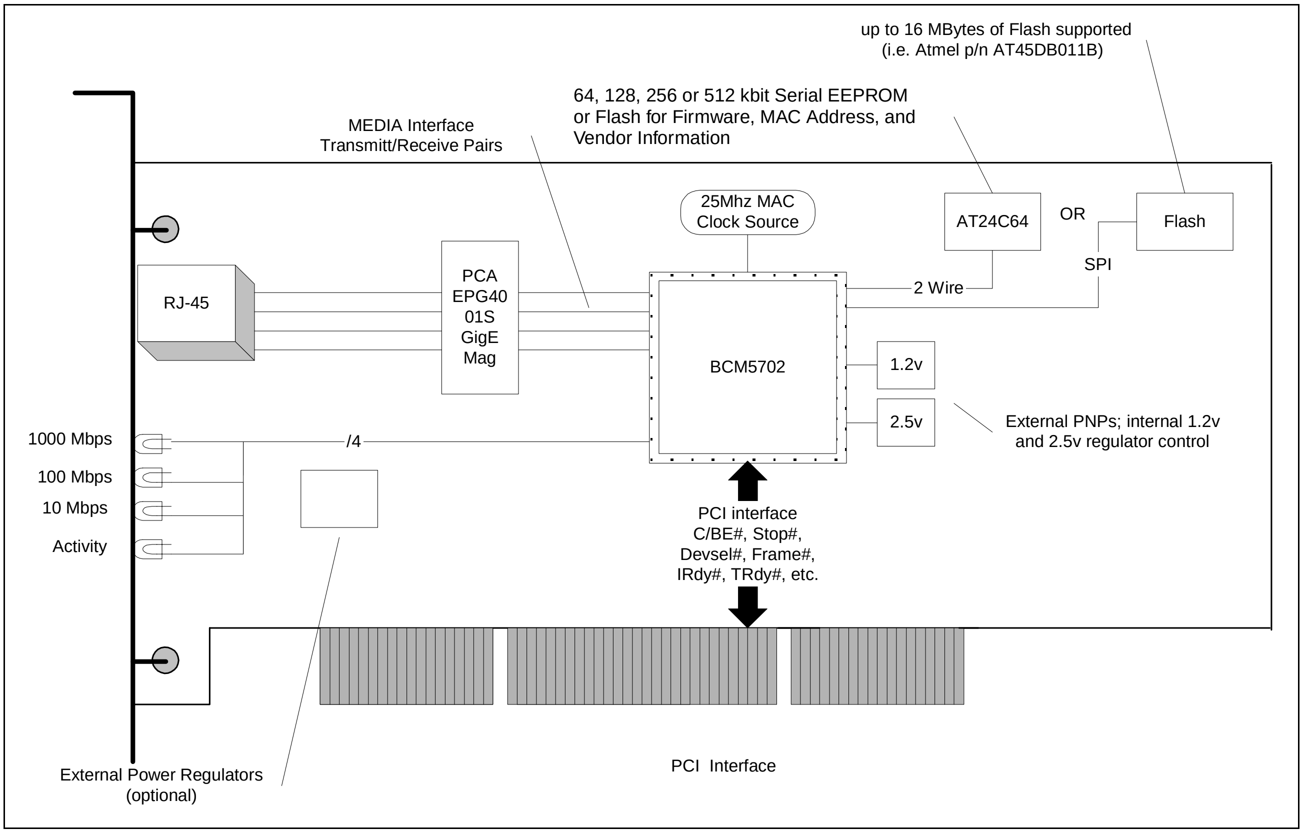


Figure 4: Typical BCM5702-Based NIC Board Block Diagram

Table 7: BCM5702 NIC Part Component Breakdown

Part Component	*Description*
RJ45	The physical connector for category 5 twisted-pair cabling.
Magnetics	The pulse H5007 isolates the physical layer from voltage events such as sags, swells, and transients. The magnetics module also compensates for impedance mismatches between the cabling and physical layer.
Clock source	A crystal oscillator generates a 25-MHz clocking signal.
Voltage regulators	Internal voltage regulator controllers provide control for external PNP controlled supplies for 2.5V and 1.2V.
MAC	The BCM5702 MAC.
PCI	The PCI specification defines a protocol for bus master controller and target data movement. The MAC is a bus master controller and may move data without CPU intervention. The BCM5702 MAC supports only the PCI v2.2 specifications.
EEPROM/Flash	The non-volatile storage for firmware boot code, MAC address, VPD, and PXE code. The EEPROM is read after the MAC is reset.

Programming Aspects

See Table 2 on page 5 for the revision levels of the BCM57XX family. Host software can use the PCI Revision ID and Chip ID information in the PCI configuration registers to determine the revision level of the BCM57XX chip on the board to load the appropriate workarounds described in the errata sheets.

Choice of host access mode determines mailbox priority (see "Configuration Space" on page 178 in Section 9: "PCI"):

- Host standard or flat mode uses the high-priority mailboxes (see "High-Priority Mailboxes" on page 370).
- Indirect mode uses the low-priority mailboxes (see "Low-Priority Mailboxes" on page 490).

See "Device Control" on page 146 for the procedure to initialize this device.

BCM5703C MAC with Integrated Transceiver

The BCM5703C device is a third-generation triple speed 10/100/1000BASE-T MAC device. The integrated 10/100/1000BASE-T Ethernet PHY device performs all the physical layer functions for 1000BASE-T, 100BASE-T, and 10BASE-T Ethernet on standard Category 5 UTP. This device is backward-compatible with BCM5701 MAC transceiver. Following are the important features of BCM5703C MAC.

- PCI-V2.2 32-bits, 33-MHz/66-MHz bus interface
- PCI-X 64-bits, 133MHz bus interface
- Dual high-speed on-chip RISC processors
- 10/100/1000BASE-T full-duplex and half-duplex MAC
- 10/100/1000BASE-T Ethernet PHY transceiver
- IEEE 802.1Q VLAN tag support
- IEEE 802.1p Layer2 priority encoding
- WOL support meeting ACPI requirements
- IEEE 802.3x flow control
- Integrated 96-KB packet buffer memory
- Programmable receive rule checker
- 9KB jumbo frame support
- Statistics for SNMP MIB II, Ethernet-like MIB, and Ethernet MIB (802.3z Clause 30)
- SMBus interface supporting ASF 1.0
- Serial EEPROM and serial Flash support
- JTAG support
- 1.2V CMOS with 5V tolerant PCI I/Os
- Available in a 400-fBGA package

Following features are the improvements over BCM5701 MAC-controller.

- Lower power consumption
- An additional 128 address hash table entries
- Support for 16 MAC-perfect filtered addresses
- Comprehensive non-volatile memory interface supporting both serial flash and serial EEPROM

TYPICAL APPLICATION

The following figure shows a typical BCM5703C-based NIC board layout.

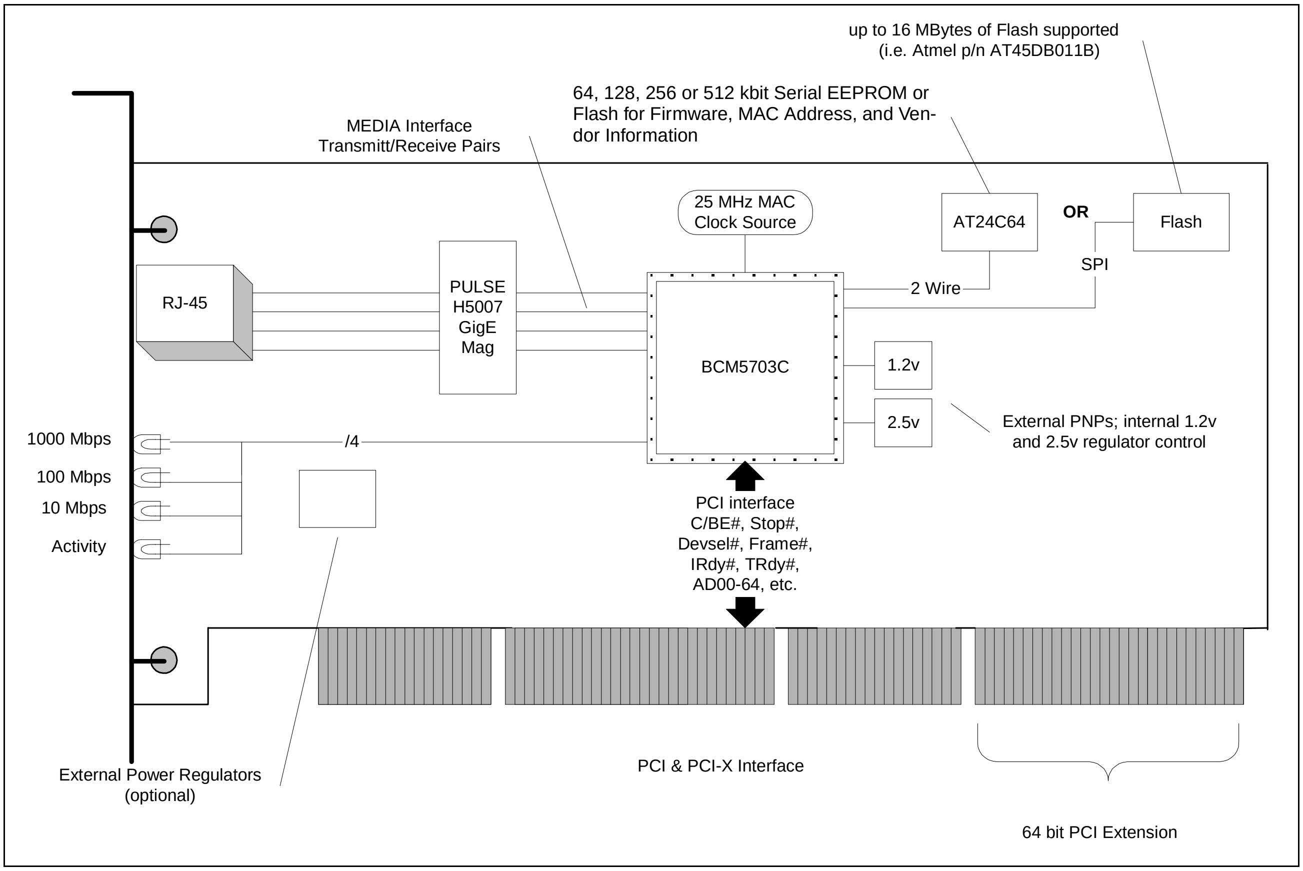

up to 16 MBytes of Flash supported
(i.e. Atmel p/n AT45DB011B)
64, 128, 256 or 512 kbit Serial EEPROM or Flash for Firmware, MAC Address, and Vendor Information
MEDIA Interface
Transmitt/Receive Pairs
25 MHz MAC Clock Source
AT24C64
OR
Flash
SPI
2 Wire
RJ-45
PULSE H5007 GigE Mag
BCM5703C
1.2v
2.5v
External PNPs; internal 1.2v and 2.5v regulator control
1000 Mbps
100 Mbps
10 Mbps
Activity
/4
PCI interface C/BE#, Stop#, Devsel#, Frame#, IRdy#, TRdy#, AD00-64, etc.
External Power Regulators (optional)
PCI & PCI-X Interface
64 bit PCI Extension

Figure 5: Typical BCM5703C-Based NIC Board Block Diagram

Table 8: BCM5703C NIC Part Component Breakdown

Part Component	***Description***
RJ45	The physical connector for category 5 twisted-pair cabling.
Magnetics	The pulse H5007 isolates the physical layer from voltage events such as sags, swells, and transients. The magnetics module also compensates for impedance mismatches between the cabling and physical layer.
Clock source	A crystal oscillator generates a 25-MHz clocking signal.
Voltage regulators	Internal voltage regulator controllers provide control for external PNP controlled supplies for 2.5V and 1.2V.
MAC	The BCM5703C MAC.
PCI/PCI-X	The PCI specification defines a protocol for bus master controller and target data movement. The MAC is a bus master controller and may move data without CPU intervention. The BCM5703C MAC supports both the PCI v2.2 and the PCI-X v1.0 specifications.
EEPROM/Flash	The non-volatile storage for firmware boot code, MAC address, VPD, and PXE code. The NVRAM is read after the MAC is reset.

PROGRAMMING ASPECTS

See Table 2 on page 5 for the revision levels of the BCM57XX family. Host software can use the PCI Revision ID and Chip ID information in the PCI configuration registers to determine the revision level of the BCM57XX chip on the board to load the appropriate workarounds described in the errata sheets.

Choice of host access mode determines mailbox priority (see "Configuration Space" on page 178 in Section 9: "PCI"):

- Host standard or flat mode uses the high-priority mailboxes (see "High-Priority Mailboxes" on page 370).
- Indirect mode uses the low-priority mailboxes (see "Low-Priority Mailboxes" on page 490).

See "Device Control" on page 146 for the procedure to initialize this device.

BCM5703S MAC WITH INTEGRATED SERDES TRANSCEIVER

The BCM5703S MAC with integrated SerDes transceiver supports a TBI for 1000BASE-SX connections. Based on Broadcom's proven DSP technology, the integrated SerDes transceiver device is a highly integrated solution combining digital adaptive equalizers, ADCs, PLLs, line drivers, echo cancellers, crosstalk cancellers, and all other required support circuitry. Following are the important features of BCM5703S device.

- PCI-V2.2 32-bits, 33-MHz/66-MHz bus interface
- PCI-X 64-bits, 133MHz bus interface
- Dual high-speed on-chip RISC processors
- 10/100/1000BASE-T full-duplex and half-duplex MAC
- 1000BASE SX SerDes transceiver
- IEEE 802.1Q VLAN tag support
- IEEE 802.1p Layer2 priority encoding
- WOL support meeting ACPI requirements
- IEEE 802.3x flow control
- Integrated 96-KB packet buffer memory
- Programmable receive rule checker
- 9KB jumbo frame support
- Statistics for SNMP MIB II, Ethernet-like MIB, and Ethernet MIB (802.3z Clause 30)
- SMBus interface supporting ASF 1.0
- Serial EEPROM and serial Flash support
- JTAG support
- 1.2V CMOS with 5V tolerant PCI I/Os
- Available in a 300 H2BGA package

TYPICAL APPLICATION

The following figure shows a typical BCM5703S-based NIC board layout.

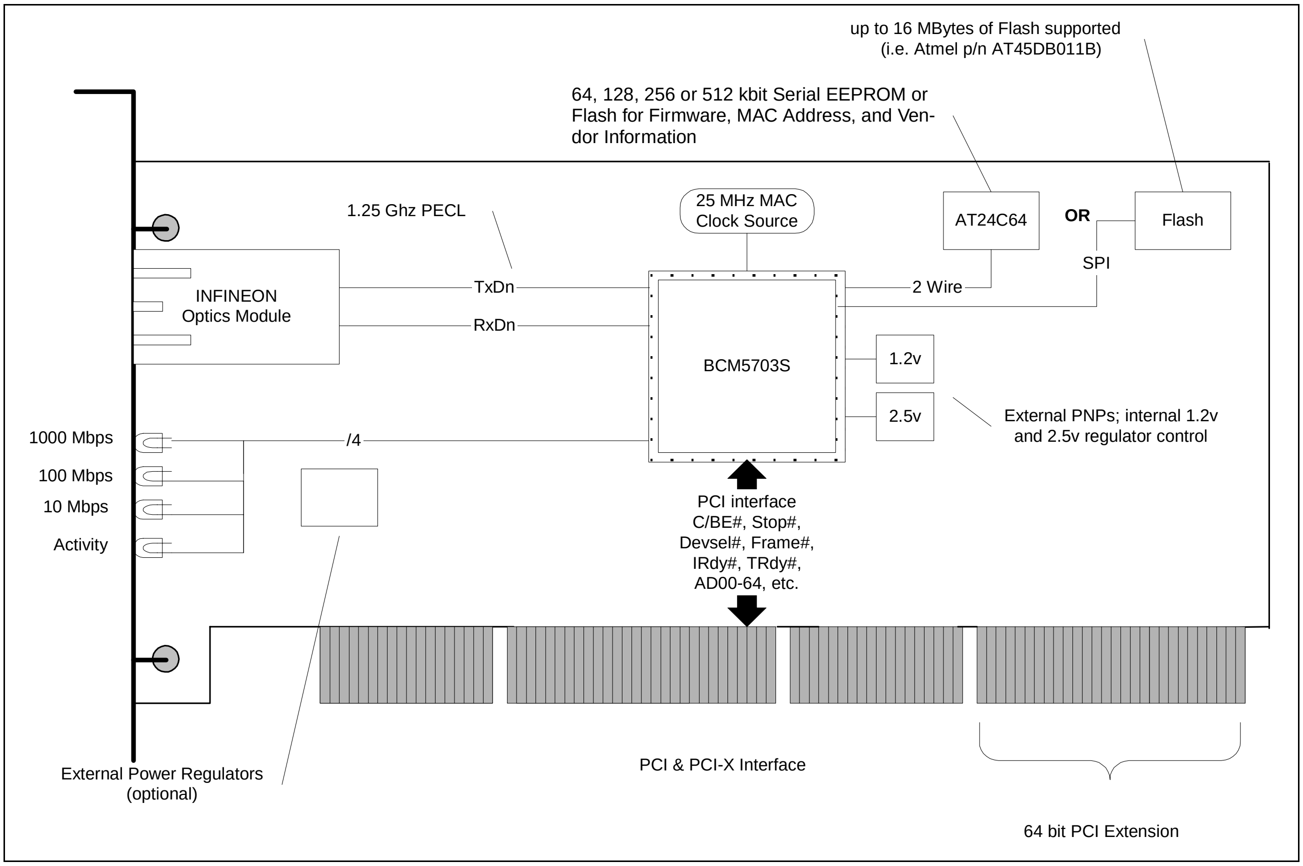

up to 16 MBytes of Flash supported
(i.e. Atmel p/n AT45DB011B)
64, 128, 256 or 512 kbit Serial EEPROM or Flash for Firmware, MAC Address, and Vendor Information
1.25 Ghz PECL
25 MHz MAC Clock Source
AT24C64
OR
Flash
SPI
INFINEON Optics Module
TxDn
RxDn
2 Wire
BCM5703S
1.2v
2.5v
External PNPs; internal 1.2v and 2.5v regulator control
1000 Mbps
100 Mbps
10 Mbps
Activity
/4
PCI interface
C/BE#, Stop#,
Devsel#, Frame#,
IRdy#, TRdy#,
AD00-64, etc.
External Power Regulators (optional)
PCI & PCI-X Interface
64 bit PCI Extension

Figure 6: Typical BCM5703S-Based NIC Board Block Diagram

Table 9: BCM5703S NIC Part Component Breakdown

Part Component	*Description*
Optics module	Multimode or single-mode fiber optic physical layer.
SerDes	The serializer-deserializer accepts TBI input and outputs a 1.25-GHz PECL output.
Clock source	Two clock sources are used in the fiber application: a 125-MHz clock for the TBI interface and a 25-MHz clock for core MAC functionality (i.e., RX, TX, PCI).
Voltage regulators	Internal voltage regulator controllers provide control for external PNP controlled supplies for 2.5V and 1.2V.
MAC	The BCM5703S MAC.
PCI/PCI-X	The PCI specification defines a protocol for bus master controller and target data movement. The MAC is a bus master controller and may move data without CPU intervention. The MAC supports both the PCI v2.2 and the PCI-X v1.0 specifications.
EEPROM/Flash	The non-volatile storage for firmware boot code, MAC address, VPD, and PXE code. The NVRAM is read after the MAC is reset.

PROGRAMMING ASPECTS

See Table 2 on page 5 for the revision levels of the BCM57XX family. Host software can use the PCI Revision ID and Chip ID information in the PCI configuration registers to determine the revision level of the BCM57XX chip on the board to load the appropriate workarounds described in the errata sheets.

Choice of host access mode determines mailbox priority (see "Configuration Space" on page 178 in Section 9: "PCI"):

- Host standard or flat mode uses the high-priority mailboxes (see "High-Priority Mailboxes" on page 370).
- Indirect mode uses the low-priority mailboxes (see "Low-Priority Mailboxes" on page 490).

See "Device Control" on page 146 for the procedure to initialize this device.

BCM5704C DUAL-MAC WITH INTEGRATED TRANSCEIVERS

The BCM5704C is a fourth generation 10/100/1000BASE-T Ethernet dual-port LAN controller designed for high density server applications. The device combines two triple-speed (1000BASE-T/100BASE-TX/10BASE-T) MACs with two triple-speed transceivers into one single monolithic CMOS chip using 0.13 micron CMOS technology. The BCM5704C device has two RISC processors and 64 KB on chip buffer memory for each function. Bridgeless arbitration architecture controls data flow between two independent PCI functions, each is independently memory-mapped and PCI configurable.

- PCI-V2.2 32-bits, 33-MHz/66-MHz bus interface
- PCI-X 32-bit/64-bit, 66/100/133MHz bus interface
- Dual high-speed on-chip RISC processors for each port (or PCI function)
- Two 10/100/1000BASE-T full-duplex and half-duplex MACs
- Two 10/100/1000BASE-T Ethernet PHY transceivers
- IEEE 802.1Q VLAN tag support
- IEEE 802.1p Layer2 priority encoding
- WOL support meeting ACPI requirements
- IEEE 802.3x flow control
- Integrated 64-KB packet buffer memory for each port
- Programmable receive rule checker
- 9KB jumbo frame support
- Statistics for SNMP MIB II, Ethernet-like MIB, and Ethernet MIB (802.3z Clause 30)
- SMBus interface supporting ASF 1.0 and IPMI v1.5
- Serial EEPROM and serial Flash support
- JTAG support
- 1.2V CMOS with 5V tolerant PCI I/Os
- Available in a 300 RBGA and 400 FBGA packages

Improvements over the third-generation integrated Gigabit Ethernet Controller include the following:

- Shared PCI/PCI-X interface bus across two internal PCI functions with separate configuration space
- Integrated dual 10/100/1000 MAC and PHY devices able to share the local bus via bridgeless arbitration
- Comprehensive non-volatile memory interface supporting both serial flash and serial EEPROM

Typical Application

The following figure shows a typical BCM5704C-based NIC board layout.

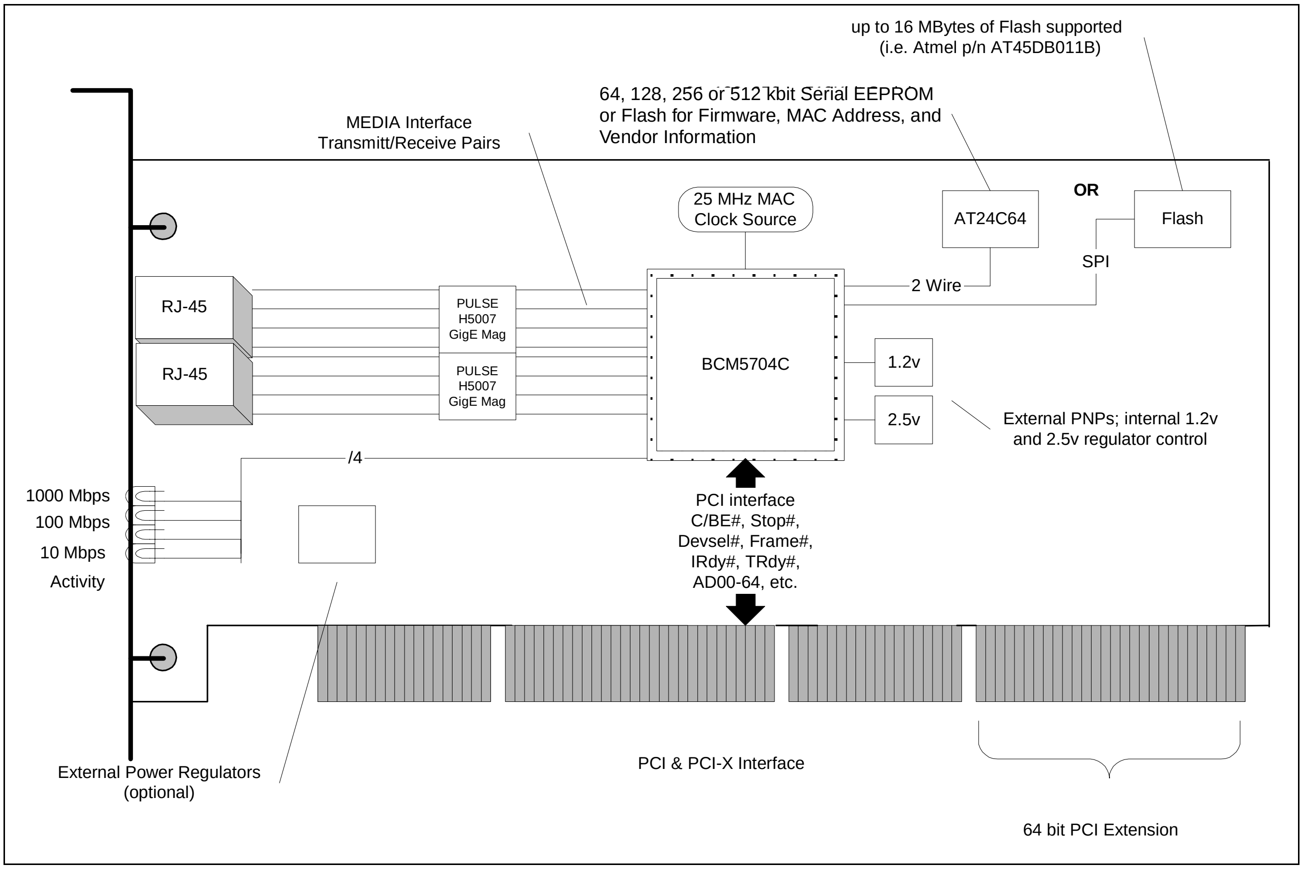


Figure 7: Typical BCM5704C-Based NIC Board Block Diagram

Table 10: BCM5704C NIC Part Component Breakdown

Part Component	Description
RJ45	The physical connector for category 5 twisted-pair cabling.
Magnetics	The pulse H5007 isolates the physical layer from voltage events such as sags, swells, and transients. The magnetics module also compensates for impedance mismatches between the cabling and physical layer.
Clock source	A crystal oscillator generates a 25-MHz clocking signal.
Voltage regulators	Internal voltage regulator controllers provide control for external PNP controlled supplies for 2.5V and 1.2V.
MAC	The BCM5704C dual MAC.
PCI/PCI-X	The PCI specification defines a protocol for bus master controller and target data movement. The MAC is a bus master controller and may move data without CPU intervention. The BCM5704C dual MAC supports both the PCI v2.2 and the PCI-X v1.0 specifications.
EEPROM/Flash	The non-volatile storage for firmware boot code, MAC address, VPD, and PXE code. The NVRAM is read after the MAC is reset.

Programming Aspects

See Table 2 on page 5 for the revision levels of the BCM57XX family. Host software can use the PCI Revision ID and Chip ID information in the PCI configuration registers to determine the revision level of the BCM57XX chip on the board to load the appropriate workarounds described in the errata sheets.

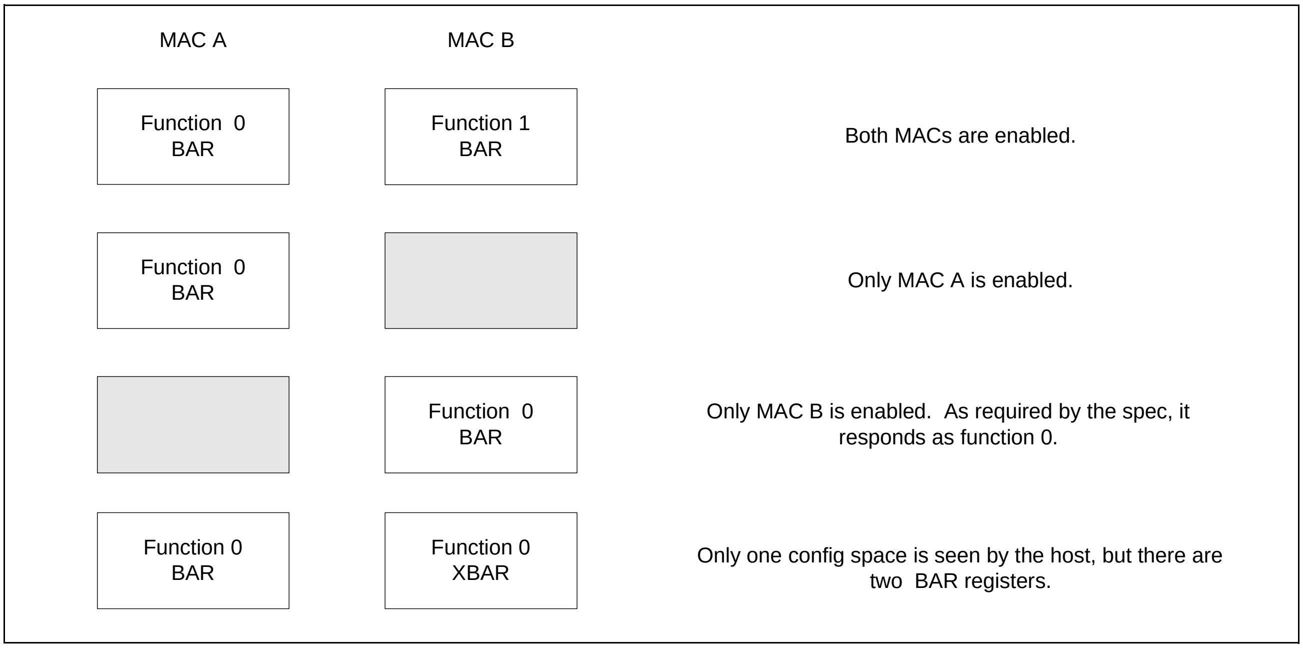
Choice of host access mode determines mailbox priority (see "Configuration Space" on page 178 in Section 9: "PCI"):

- Host standard or flat mode uses the high-priority mailboxes (see "High-Priority Mailboxes" on page 370).
- Indirect mode uses the low-priority mailboxes (see "Low-Priority Mailboxes" on page 490).

Dual MAC Modes of Operation

In default mode, BCM5704C dual MAC transceiver is considered as two independent devices. The host program can configure it to enable only one device or to configure two devices as one for teaming.

In the two-function mode, each MAC has its own PCI configuration space registers. Within this space, there is the Base Address register and the PCI address range which the MAC registers reside. The BCM5704C dual-MAC transceiver can operate in the four different modes shown in Figure 8.



Both MACs are enabled.

Only MAC A is enabled.

Only MAC B is enabled. As required by the spec, it responds as function 0.

Only one config space is seen by the host, but there are two BAR registers.

Figure 8: Dual MAC Modes of Operation

The following table describes the dual MAC modes.

Table 11: Dual MAC Modes of Operation

Mode	Description
Both MACs enabled	This is the state of the BCM5704 in normal operation. Each MAC has its own config space and independent BAR registers.
MAC A only enabled	If MAC B is disabled, it does not affect the operation of MAC A, which continues to respond to config cycles to function 0.
MAC B only enabled	Since the PCI spec requires all devices to contain a function 0, MAC B will respond to function 0 config cycles.
BAR-XBAR mode	Only MAC A responds to config cycles, but MAC A contains two Base Address registers. The second Base Address register (XBAR) (see "MAC 0 XBAR Register (Offset 0x18)" on page 307) is the location in PCI space where MAC B registers reside. In this state, MAC B is still controlled by the MAC B config registers, which are not accessible through config space. So BCM5704C software will be required to copy MAC A config registers into MAC B config space through memory write cycles.

The boot code reads NVRAM location 0xC8 (i.e., H/W Configuration) for dual MAC mode configuration and programs the Dual MAC Control register (see "Dual-MAC Control Register (Offset 0xB8)" on page 346). The Dual MAC Control register exists in both MACs and must be written to the same value.

Table 12: Dual-MAC Control Register Channel Control Bits

Value	Mode
00	Both MACs enabled
01	MAC B only
10	MAC A only
11	BAR-XBAR mode

GPIO

The BCM5703C MAC-transceiver contains three GPIO bits that can be controlled or observed through PCI registers. The BCM5704C dual MAC-transceiver also contains three GPIO bits, but each MAC contains a set of registers that control these bits. In order to let each of the MACs control the GPIO bits, use the following rules:

- If neither version of the GPIO_OE is enabled, the GPIO bit is tristated.
- If only one version of the GPIO_OE is enabled, the GPIO bit is driven with the value in the corresponding GPIO_DATA bit.
- If both versions of the GPIO_OE are enabled and the GPIO_DATA bits are the same, that value is driven on the GPIO bit.
- If both versions of the GPIO_OE are enabled and the GPIO_DATA bits are different, the GPIO bit is tristated.

WOL

The GPIO0, GPIO1, and GPIO2 are used to switch between Vdd and Vaux. In the BCM5704C/BCM5704S dual MAC-Controller, the shared GPIOs can be used for the same purpose by following these rules:

- On reset, OE is inactive.
- When the device is loaded, if WOL is enabled, enable OE and drive OUT low.
- When the device is loaded, if WOL is disabled, leave OE inactive.
- On WOL, drive OUT high.

To support the case where one device is not loaded, MAC A and MAC B must handshake to ensure both are ready for low-power mode before performing the GPIO write that will switch the power source (see the following figure).

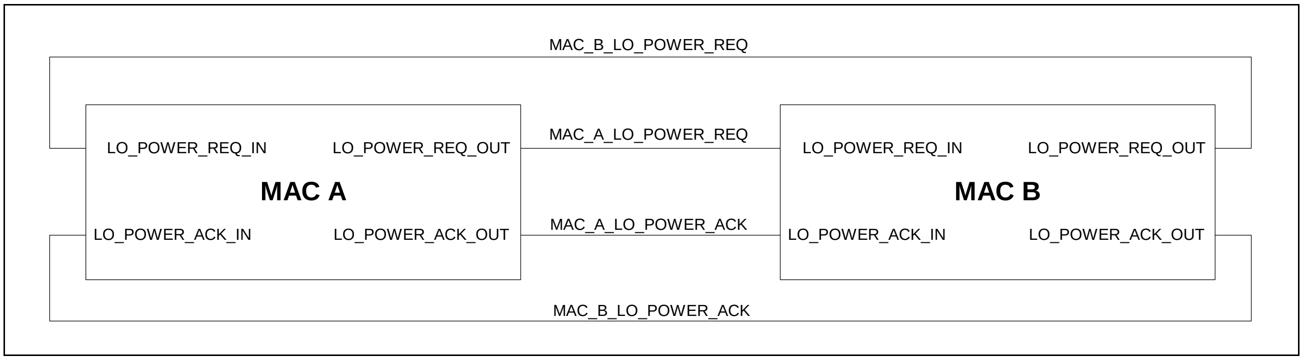


Figure 9: Handshaking Using Power Signal Status Signals

The LOW__POWER_ACK signals initialize inactive. If a port is disabled, LOW_POWER_ACK will be driven high. When the first MAC receives the low-power event, it asserts its LOW_POWER_REQ and LOW_POWER_ACK. If the other MAC has a device loaded, it will wait until receiving a low-power event and then assert its LOW_POWER_REQ and LOW_POWER_ACK. If the other MAC has no device loaded, it will go into low-power mode and then assert LOW_POWER_ACK.

A MAC that is asserting LOW_POWER_REQ does not perform the GPIO write until LOW_POWER_ACK is returned active. This guarantees that we do not switch the Vaux until both MACs are in low-power mode.

$\overline{\text{PME}}$, $\overline{\text{INTA}}$

Each MAC has a unique version of $\overline{\text{PME}}$ which are logically ORed to generate $\overline{\text{PME}}$ on the PCI bus. Each MAC has a unique version of $\overline{\text{INTA}}$. The MAC A $\overline{\text{INTA}}$ is mapped to $\overline{\text{INTA}}$. The MAC B $\overline{\text{INTA}}$ is mapped to $\overline{\text{INTB}}$.

NVRAM

NVRAM accesses use the same software arbitration as the BCM5703C MAC-transceiver. The arbiters are chained together as shown in the following figure.

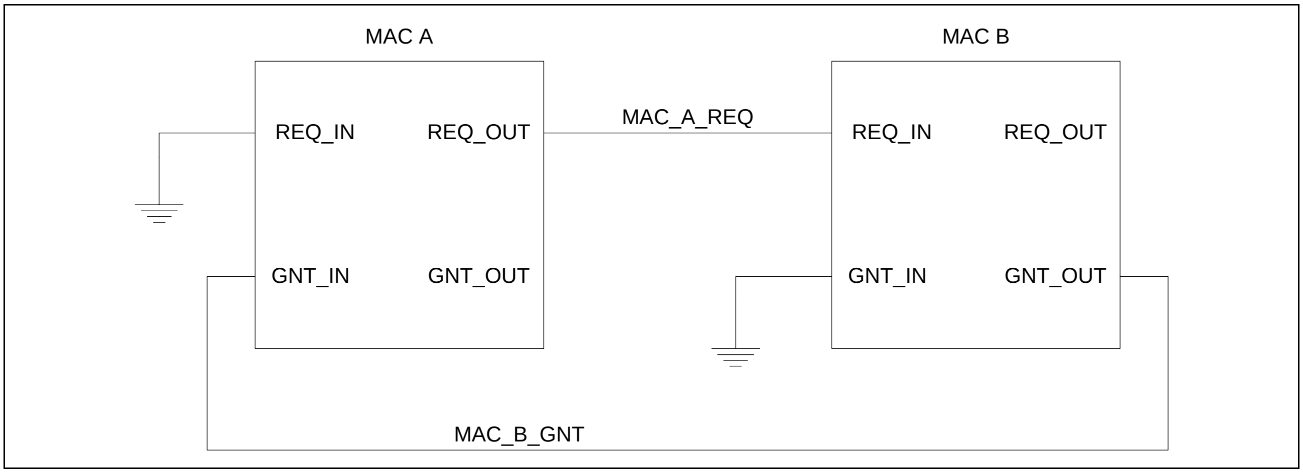


Figure 10: Daisy-Chained Arbiters

REQ_OUT is the OR of the local requests. GNT_OUT is the OR of the local grants. MAC A has higher priority, as MAC_A_REQ will block a MAC B request. Once MAC B has been granted the NVRAM, MAC_B_GNT prevents MAC A from being granted the bus until MAC B has finished.

The single MAC NVRAM Arbitration is implemented through the Software Arbitration register (see "Software Arbitration Register (Offset 0x7020)" on page 555). The following table defines the register bits.

Table 13: Software Arbitration Register Bits

Bit	*Name*	*Description*
3:0	SET_REQ	When this bit is written high, the corresponding request bit is set.
7:4	CLR_REQ	When this bit is written high, the corresponding request bit is cleared.
11:8	ARB_WON	When this bit is read high, the corresponding requestor controls the NVRAM. Once this bit is high, it will remain high until CLR_REQ is written high.
15:12	SW_REQ	This bit reflects the value of the corresponding request.

The arbitration is a simple priority scheme with bit 0 being the highest priority and bit 3 lowest priority.

Initialization

See "Device Control" on page 146 for the procedure to initialize this device.

BCM5704S DUAL-MAC WITH INTEGRATED SERDES TRANSCEIVERS

The BCM5704S is a fourth generation device that combines two triple-speed (1000BASE-T/100BASE-TX/10BASE-T) MACs with two 1000BASE-X SerDes transceivers into one single monolithic CMOS chip using 0.13 micron CMOS technology. The BCM5704S device has two RISC processors and 64 KB on chip buffer memory for each function. Bridgeless arbitration architecture controls data flow between two independent PCI functions, each is independently memory-mapped and PCI configurable.

- PCI-V2.2 32-bits, 33-MHz/66-MHz bus interface
- PCI-X 32-bit/64-bit, 66/100/133MHz bus interface
- Dual high-speed on-chip RISC processors for each port (or PCI function)
- Two 10/100/1000BASE-T full-duplex and half-duplex MACs
- Two 1000BASE-X SerDes transceivers
- IEEE 802.1Q VLAN tag support
- IEEE 802.1p Layer2 priority encoding
- WOL support meeting ACPI requirements
- IEEE 802.3x flow control
- Integrated 64-KB packet buffer memory for each port
- Programmable receive rule checker
- 9KB jumbo frame support
- Statistics for SNMP MIB II, Ethernet-like MIB, and Ethernet MIB (802.3z Clause 30)
- SMBus interface supporting ASF 1.0 and IPMI v1.5
- Serial EEPROM and serial Flash support
- JTAG support
- 1.2V CMOS with 5V tolerant PCI I/Os
- Available in a 300 RBGA and 400 FBGA packages

Improvements over the third-generation integrated Gigabit Ethernet Controller include the following:

- Shared PCI/PCI-X interface bus across two internal PCI functions with separate configuration space
- Integrated dual 10/100/1000 MAC and PHY devices able to share the local bus via bridgeless arbitration
- Comprehensive non-volatile memory interface supporting both serial flash and serial EEPROM

Typical Application

The following figure shows a typical BCM5704S-based NIC board layout.

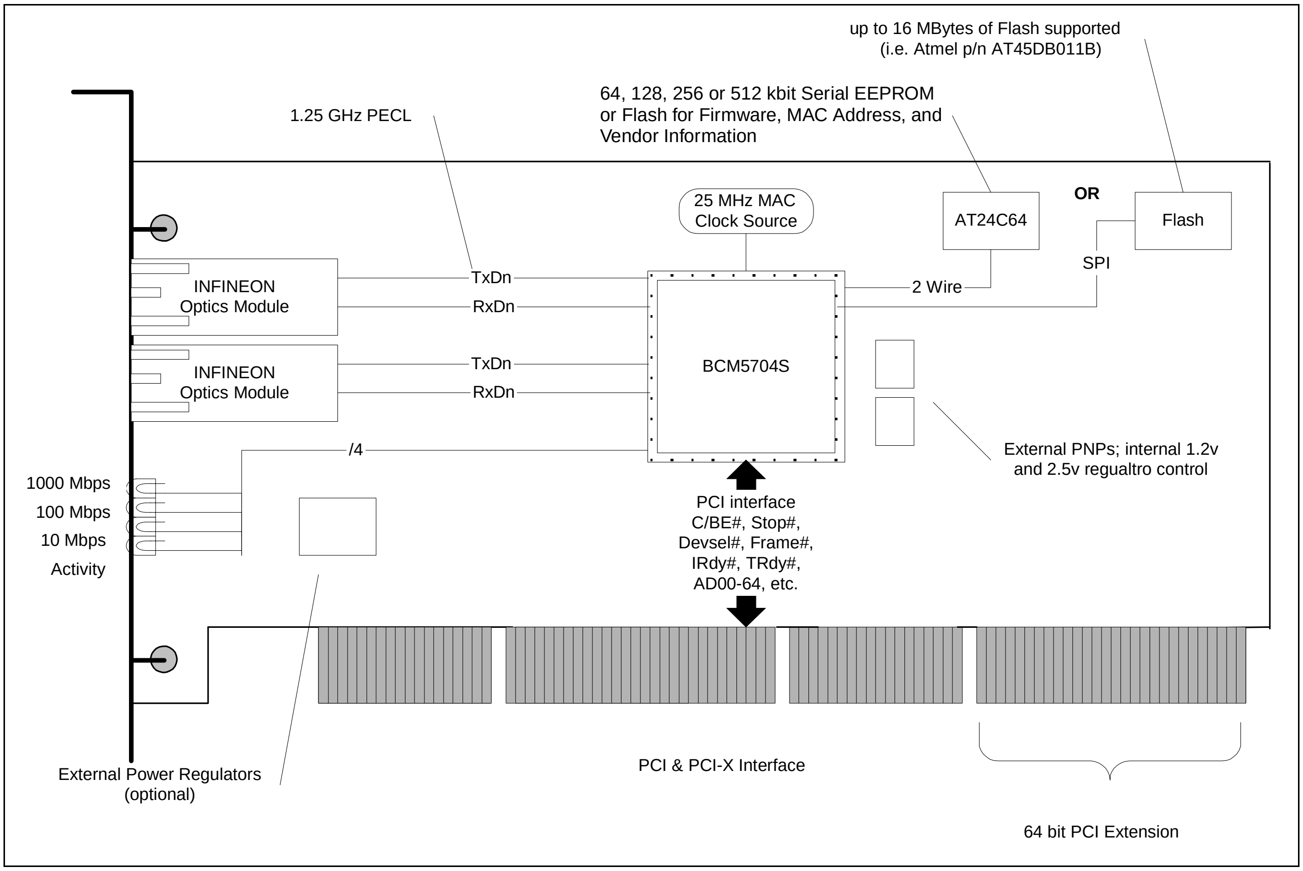


Figure 11: Typical BCM5704S-Based NIC Board Block Diagram

Table 14: BCM5704S NIC Part Component Breakdown

Part Component	*Description*
Optics module	Multimode or single-mode fiber optic physical layer.
SerDes	The serializer-deserializer accepts TBI input and outputs a 1.25-GHz PECL output.
Clock source	Two clock sources are used in the fiber application: a 125-MHz clock for the TBI interface and a 25 MHz clock for core MAC functionality (i.e., RX, TX, PCI).
Voltage regulators	Internal voltage regulator controllers provide control for external PNP controlled supplies for 2.5V and 1.2V.
MAC	The BCM5704S dual MAC.
PCI/PCI-X	The PCI specification defines a protocol for bus master controller and target data movement. The MAC is a bus master controller and may move data without CPU intervention. The BCM5704S MAC supports both the PCI v2.2 and the PCI-X v1.0 specifications.
EEPROM/Flash	The non-volatile storage for firmware boot code, MAC address, VPD, and PXE code. The NVRAM is read after the BCM5704S dual MAC transceiver SerDes is reset.

PROGRAMMING ASPECTS

See Table 2 on page 5 for the revision levels of the BCM57XX family. Host software can use the PCI Revision ID and Chip ID information in the PCI configuration registers to determine the revision level of the BCM57XX chip on the board to load the appropriate workarounds described in the errata sheets.

Choice of host access mode determines mailbox priority (see "Configuration Space" on page 178 in Section 9: "PCI"):

- Host standard or flat mode uses the high-priority mailboxes (see "High-Priority Mailboxes" on page 370).
- Indirect mode uses the low-priority mailboxes (see "Low-Priority Mailboxes" on page 490).

Dual MAC Modes of Operation

This section for the BCM5704S dual-MAC transceiver SerDes is the same as "Dual MAC Modes of Operation" on page 25 for the BCM5704C dual-MAC transceiver.

GPIO

This section for the BCM5704S dual-MAC transceiver SerDes is the same as "GPIO" on page 26 for the BCM5704C dual-MAC transceiver.

WOL

This section for the BCM5704S dual-MAC transceiver SerDes is the same as "WOL" on page 27 for the BCM5704C dual-MAC transceiver.

$\overline{\text{PME}}$, $\overline{\text{INTA}}$

This section for the BCM5704S dual-MAC transceiver SerDes is the same as "PME, INTA" on page 27 for the BCM5704C dual-MAC transceiver.

NVRAM

This section for the BCM5704S dual-MAC transceiver SerDes is the same as "NVRAM" on page 28 for the BCM5704C dual-MAC transceiver.

Initialization

See "Device Control" on page 146 for the procedure to initialize this device.

BCM5705 AND BCM5788 MACS WITH INTEGRATED TRANSCEIVERS

The BCM5705 MAC is a fourth generation device that includes a triple speed MAC and a 10/100/1000BASE-T Ethernet transceiver. The device is suited for applications such as low-cost NIC, low cost LAN on Motherboard (LOM), and other applications that support the CAT 5 connection. Following are the important features of BCM5705 and BCM5788 devices. The BCM5788 does not support 64-bit addressing capability and DAC capability.

- PCI-V2.3 32-bits, 33-MHz/66-MHz bus interface
- Single high-speed on-chip RISC Processor
- 10BT/100BTX/1000BT full-duplex and half-duplex MAC
- 10BT/100BTX/1000BT Ethernet PHY transceiver
- IEEE 802.1Q VLAN tag support
- IEEE 802.1p Layer2 priority encoding
- WOL support meeting ACPI requirements
- IEEE 802.3x flow control
- Integrated 56-KB Rx packet buffer memory and 8-KB Tx packet buffer memory
- Programmable receive rule checker
- SMBus interface supporting ASF 1.0
- Serial EEPROM and serial Flash support
- JTAG support
- 1.2V CMOS with 5V tolerant PCI I/Os
- Available in a 196 FBGA package

The BCM5705 MAC-transceiver is identical to the BCM5703C MAC transceiver except:

- Only PCI-v2.3 32-bits, 33-MHz/66-MHz bus interface
- Single internal CPU (RX CPU)
- 56 KB RXMBUF with configurable partition between receive data buffer and CPU scratchpad (one common configuration is 32 KB for receive frame buffer and 24 KB for CPU scratchpad memory)
- Separate 8 KB TXMBUF to store transmit data packets
- Single send ring
- Single receive return ring
- Standard receive producer ring only
- Only four receive rules registers
- No jumbo frame support
- Host-based send BD only

Lower power state for NIC or mobile applications**TYPICAL APPLICATION**

The following figure shows a typical BCM5705-based NIC board layout.

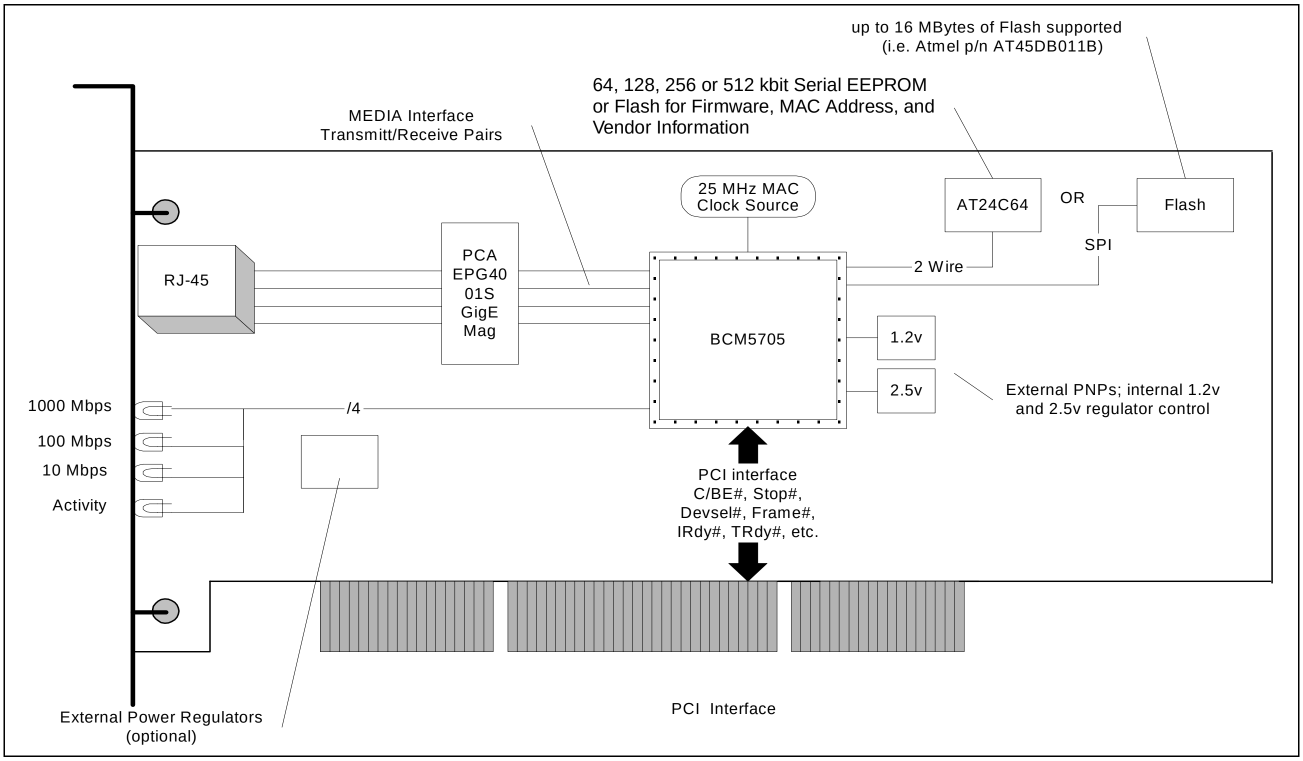


Figure 12: Typical BCM5705-Based NIC Board Block Diagram

Table 15: BCM5705 NIC Part Component Breakdown

Part Component	***Description***
RJ45	The physical connector for category 5 twisted-pair cabling.
Magnetics	The pulse H5007 isolates the physical layer from voltage events such as sags, swells, and transients. The magnetics module also compensates for impedance mismatches between the cabling and physical layer.
Clock source	A crystal oscillator generates a 25-MHz clocking signal.
Voltage regulators	Internal voltage regulator controllers provide control for external PNP controlled supplies for 2.5V and 1.2V.
MAC	The BCM5705 MAC.
PCI	The PCI specification defines a protocol for bus master controller and target data movement. The MAC is a bus master controller and may move data without CPU intervention. The BCM5705 MAC supports only the PCI v2.3 specification.
EEPROM/Flash	The non-volatile storage for firmware boot code, MAC address, VPD, and PXE code. The NVRAM is read after the BCM5705 MAC transceiver is reset.

Programming Aspects

See Table 2 on page 5 for the revision levels of the BCM57XX family. Host software can use the PCI Revision ID and Chip ID information in the PCI configuration registers to determine the revision level of the BCM57XX chip on the board to load the appropriate workarounds described in the errata sheets.

Choice of host access mode determines mailbox priority (see "Configuration Space" on page 178 in Section 9: "PCI"):

- Host standard or flat mode uses the high-priority mailboxes (see "High-Priority Mailboxes" on page 370).
- Indirect mode uses the low-priority mailboxes (see "Low-Priority Mailboxes" on page 490).

See "Device Control" on page 146 for the procedure to initialize this device.

BCM5721 MAC with Integrated Transceiver

The BCM5721 is a seventh-generation 10/100/1000BASE-T Ethernet LAN controller solution for high-performance server network applications. The device combines a triple-speed IEEE 802.3 compliant MAC with a triple-speed Ethernet transceiver, a 1x PCIe bus interface, and on-chip buffer memory in a single device. Following are important features of BCM5721 device.

- PCIe v1.0a, x1 link width interface
- Single high-speed on-chip RISC Processor
- 10BT/100BTX/1000BT full-duplex and half-duplex MAC
- 10BT/100BTX/1000BT Ethernet PHY transceiver
- IEEE 802.1Q VLAN tag support
- IEEE 802.1p Layer2 priority encoding
- WOL support meeting ACPI requirements
- TCP Segmentation support
- IEEE 802.3x flow control
- Integrated 64-KB Rx packet buffer memory and 8-KB Tx packet buffer memory
- Programmable receive rule checker
- SMBus interface supporting ASF 2.0 and IPMI v1.5
- Failover and Teaming capabilities
- Serial EEPROM and serial Flash support
- JTAG support
- 1.2V CMOS with 5V tolerant PCI I/Os
- Available in a 196 FBGA and 400 FBGA packages

The BCM5721 MAC-transceiver is identical to the BCM5703C MAC transceiver except:

- PCIe v1.0a, x1 link width interface
- Single internal CPU (RX CPU)
- 64 KB RXMBUF with configurable partition between receive data buffer and CPU scratchpad
- 8 KB TXMBUF to store transmit data packets
- Single send ring
- Single receive return ring
- Standard receive producer ring only

- Only four receive rules registers
- No KB jumbo frame support
- Host base send BD only
- Lower power state for NIC or mobile applications

Typical Application

The following figure shows a typical BCM5721-based NIC board layout.

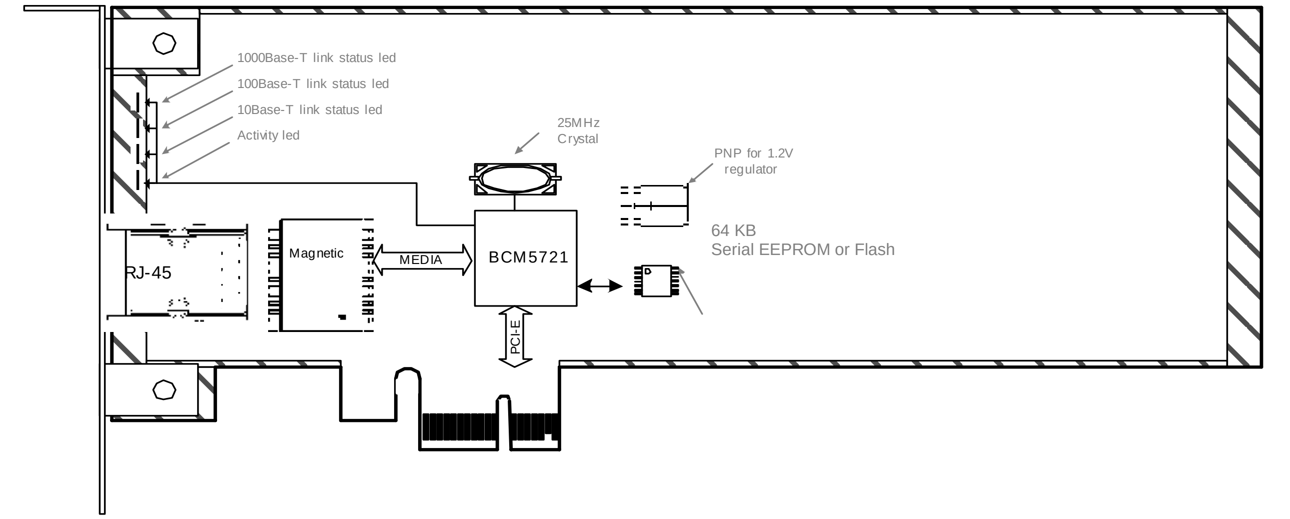


Figure 13: Typical BCM5721-Based NIC Board Block Diagram

Table 16: BCM5721 NIC Part Component Breakdown

Part Component	*Description*
RJ45	The physical connector for category 5 twisted-pair cabling.
Magnetics	The pulse H5007 isolates the physical layer from voltage events such as sags, swells, and transients. The magnetics module also compensates for impedance mismatches between the cabling and physical layer.
Clock source	A crystal oscillator generates a 25-MHz clocking signal.
Voltage regulators	Internal voltage regulator controllers provide control for external PNP controlled supplies for 2.5V and 1.2V.
MAC	The BCM5721 MAC.
PCIe	This specification defines a protocol for bus master controller and target data movement. The MAC is a bus master controller and may move data without CPU intervention.
EEPROM/Flash	The non-volatile storage for firmware boot code, MAC address, VPD, and PXE code. The NVRAM is read after the MAC is reset.

PROGRAMMING ASPECTS

See Table 2 on page 5 for the revision levels of the BCM57XX family. Host software can use the PCI Revision ID and Chip ID information in the PCI configuration registers to determine the revision level of the BCM57XX chip on the board to load the appropriate workarounds described in the errata sheets.

Choice of host access mode determines mailbox priority (see "Configuration Space" on page 178 in Section 9: "PCI"):

- Host standard or flat mode uses the high-priority mailboxes (see "High-Priority Mailboxes" on page 370).
- Indirect mode uses the low-priority mailboxes (see "Low-Priority Mailboxes" on page 490).

See "Device Control" on page 146 for the procedure to initialize this device.

BCM5751 MAC WITH INTEGRATED TRANSCEIVER

The BCM5751 is a fifth-generation 10/100/1000BASE-T Ethernet LAN controller solution for high-performance network applications. The device combines a triple-speed IEEE 802.3 compliant MAC with a triple-speed Ethernet transceiver, a 1x PCIe bus interface, and on-chip buffer memory in a single device. Following are the important features of BCM5751 device.

- PCIe v1.0a, x1 link width interface
- Single high-speed on-chip RISC processor
- 10BT/100BTX/1000BT full-duplex and half-duplex MAC
- 10BT/100BTX/1000BT Ethernet PHY transceiver
- IEEE 802.1Q VLAN tag support
- IEEE 802.1p Layer2 priority encoding
- WOL support meeting ACPI requirements
- TCP segmentation support
- IEEE 802.3x flow control
- Integrated 64-KB Rx packet buffer memory and 8-KB Tx packet buffer memory
- Programmable receive rule checker
- SMBus interface supporting ASF 2.0
- Failover and teaming capabilities
- Serial EEPROM and serial Flash support
- JTAG support
- 1.2V CMOS with 5V tolerant PCI I/Os
- Available in a 196 FPBGA package

The BCM5751 MAC transceiver is identical to the BCM5703C MAC transceiver except:

- PCIe v1.0a, x1 link width interface
- Single internal CPU (RX CPU)
- 64 KB RXMBUF with configurable partition between receive data buffer and CPU scratchpad
- 8 KB TXMBUF to store transmit data packets
- Single send ring
- Single receive return ring
- Standard receive producer ring only

- Only four receive rules registers
- No jumbo frame support
- Host-based send BD only
- Lower power state for NIC or mobile application

The BCM5751T and BCM5751TM MACs also have an integrated Trusted Platform Module (TPM) Security Processor.

TYPICAL APPLICATION

The following figure shows a typical BCM5751-based NIC board layout.

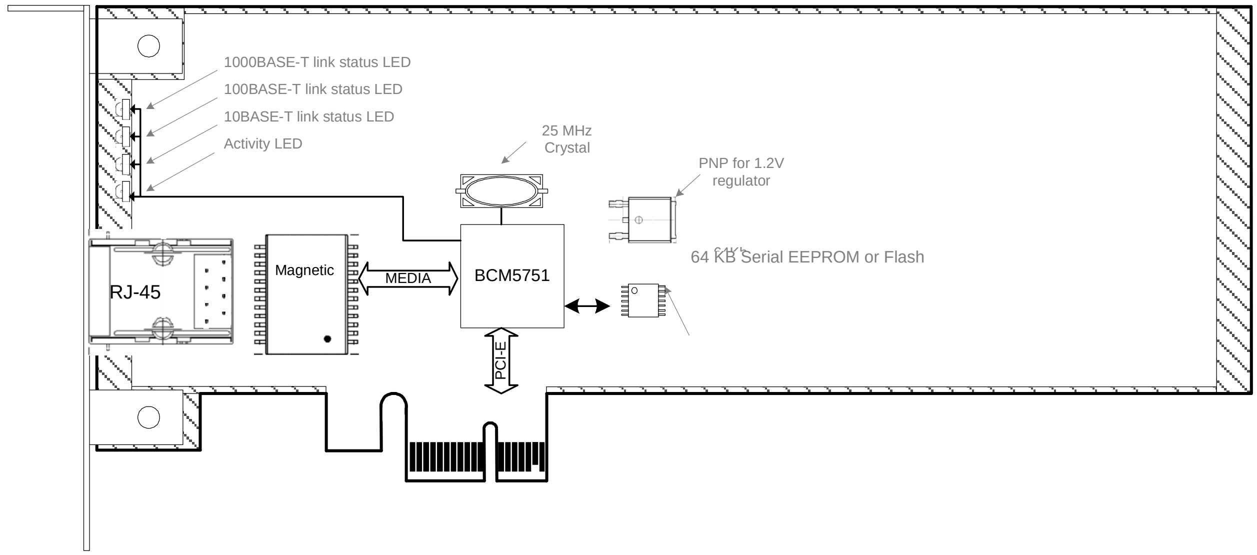


Figure 14: Typical BCM5751-Based NIC Board Block Diagram

Table 17: BCM5751 NIC Part Component Breakdown

Part Component	*Description*
RJ45	The physical connector for category 5 twisted-pair cabling.
Magnetics	The pulse H5007 isolates the physical layer from voltage events such as sags, swells, and transients. The magnetics module also compensates for impedance mismatches between the cabling and physical layer.
Clock source	A crystal oscillator generates a 25-MHz clocking signal.
Voltage regulators	Internal voltage regulator controllers provide control for external PNP controlled supplies for 2.5V and 1.2V.
MAC	The BCM5721 MAC.
PCIe	This specification defines a protocol for bus master controller and target data movement. The MAC is a bus master controller and may move data without CPU intervention.
EEPROM/Flash	The non-volatile storage for firmware boot code, MAC address, VPD, and PXE code. The NVRAM is read after the MAC is reset.

Programming Aspects

See Table 2 on page 5 for the revision levels of the BCM57XX family. Host software can use the PCI Revision ID and Chip ID information in the PCI configuration registers to determine the revision level of the BCM57XX chip on the board to load the appropriate workarounds described in the errata sheets.

Choice of host access mode determines mailbox priority (see "Configuration Space" on page 178 in Section 9: "PCI"):

- Host standard or flat mode uses the high-priority mailboxes (see "High-Priority Mailboxes" on page 370).
- Indirect mode uses the low-priority mailboxes (see "Low-Priority Mailboxes" on page 490).

See "Device Control" on page 146 for the procedure to initialize this device.

BCM5714C Dual-MAC Chip with Integrated Transceiver

The BCM5714C is a fourth-generation 10/100/1000BASE-T dual-port LAN controller for high-performance server applications. The device combines dual triple-speed IEEE 802.3 compliant MACs with dual 10/100/1000 Ethernet transceivers (PHY), a PCIe to PCI-X Internal bridge for connecting to the two MACs, a PCI-X Internal to PCI-X bridge for supporting PCI/PCI-X devices downstream, a UMP, and on-chip Frame Buffer Memory in a single device. The device is fabricated in a 1.2V CMOS process providing a low-power system solution.

The BCM5714C provides a single x4 PCIe host interface and supports one PCIe to PCI-X Internal Bridge function, two Gigabit Ethernet Controller functions, and one PCI-X Internal to PCI-X v1.0 Bridge function. Each function has its own PCI configuration space. The PCI-X Internal to PCI-X 1.0 bridge supports a 64-bit bus operating at 133 MHz, 100 MHz, 66 MHz, or 33 MHz. In PCI mode of this bridge, the bus is compliant with the PCI V2.2 specification. Each port has a dedicated on-chip high-performance RISC processor for custom frame processing. The BCM5714C also supports a UMP interface for high-speed system management traffic. Following are the important features of BCM5714C device.

- Dual 10BASE-T/100BASE-TX/1000BASE-T full-duplex/half-duplex MACs
- Dual 10BASE-T/100BASE-TX/1000BASE-T Ethernet PHY transceivers
- x4 PCIe host interface with x4 PCIe to PCI-X Internal Bridge for connecting to dual MACs and PCI-X Internal to PCI-X v1.0 Bridge.
- PCI-X Internal to PCI-X v1.0 Internal Bridge with PCI-X v1.0 secondary interface supporting 32-bit/64-bit, 66-MHz/100-MHz/133-MHz bus interface and PCI v2.2 32-bit/64-bit, 33-MHz/66-MHz
- UMP
- TCP segmentation, IP fragmentation, and reassembly
- IEEE 802.1Q VLAN tag support
- IEEE 802.1p Layer2 priority encoding
- IEEE 802.3x flow control
- CPU offload functions: TCP, IP, UDP checksum
- 32 KB dedicated receive buffer
- 22 KB transmit buffer to store TX packets and IPMI packets
- Jumbo frame support (9 KB)
- ASF 2.0 support
- IPMI Pass-through support
- ACPI 1.1 compliant
- 36 KB of scratch pad memory

- One send ring, one receive return ring, and one standard receive producer ring
- SMBus 2.0 controller
- Out-of-Box WOL support
- PXE 2.0 Remote Boot support
- 484 PBGA Package

TYPICAL APPLICATION

The following figure shows a typical BCM5714C-based LOM design.

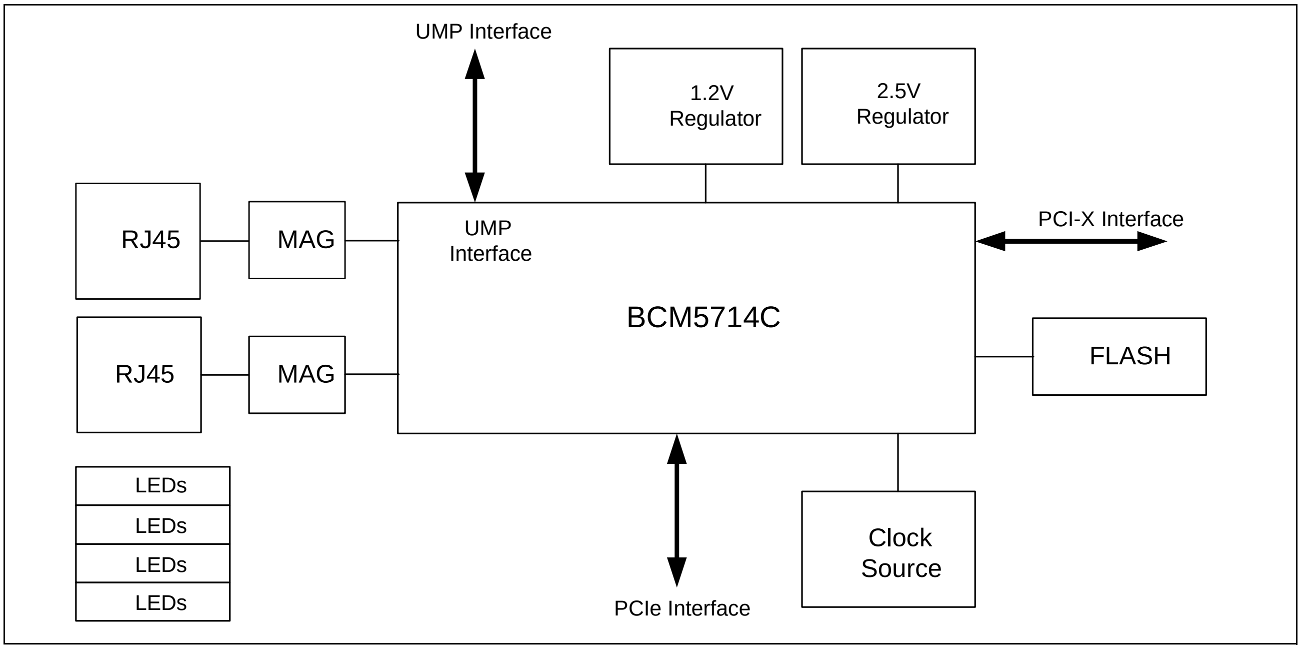


Figure 15: Typical BCM5714C-Based LOM Design Block Diagram

Table 18: BCM5714C NIC Part Component Breakdown

Part Component	*Description*
RJ45	The physical connector for category 5 twisted-pair cabling.
Magnetics	The pulse H5007 isolates the physical layer from voltage events such as sags, swells, and transients. The magnetics module also compensates for impedance mismatches between the cabling and physical layer.
Clock source	A crystal oscillator generates a 25-MHz clocking signal.
Voltage regulators	Internal voltage regulator controllers provide control for external PNP controlled supplies for 2.5V and 1.2V.
MAC	The BCM5714C MAC.
EEPROM/Flash	The non-volatile storage for firmware boot code, MAC address, VPD, and PXE code. The NVRAM is read after the MAC is reset.

BCM5714S DUAL-MAC CHIP WITH INTEGRATED FIBER TRANSCEIVER

The BCM5714S is a fourth-generation 1000BASE-X dual-port LAN controller for high performance server applications. The device combines dual triple-speed IEEE 802.3 compliant MACs with dual SerDes transceivers, a PCIe to PCI-X Internal bridge, PCI-X Internal to PCI-X v1.0 Bridge, a UMP and an on-chip memory buffer in a single device. The device is fabricated in a 1.2V CMOS process providing a low-power system solution.

The BCM5714S device is same as BCM5714C device except that it has two integrated SerDes transceivers instead of two 10/100/1000BASE-T transceivers. Following are the important features of BCM5714S device.

- Dual 10BASE-T/100BASE-TX/1000BASE-T full-duplex/half-duplex MACs
- Dual 1000BASE-X Ethernet SerDes transceiver
- x4 PCIe host interface with x4 PCIe to PCI-X Internal Bridge for connecting to dual MACs and PCI-X Internal to PCI-X v1.0 Bridge.
- PCI-X Internal to PCI-X v1.0 Internal Bridge with PCI-X v1.0 secondary interface supporting 32-bit/64-bit, 66-MHz/100-MHz/133-MHz bus interface and PCI v2.2 32/64-bit, 66-MHz/100-MHz/133-MHz
- UMP
- TCP segmentation, IP fragmentation and reassembly
- IEEE 802.1Q VLAN tag support
- IEEE 802.1p Layer2 priority encoding
- IEEE 802.3x flow control
- CPU offload functions: TCP, IP, UDP checksum
- 32 KB dedicated receive buffer
- 22 KB transmit buffer to store TX packets and IPMI packets
- Jumbo frame support (9 KB)
- ASF 2.0 support
- IPMI Pass-through support
- ACPI 1.1 compliant
- 36 KB of scratch pad memory
- One send ring, one receive return ring, and one standard receive producer ring
- SMBus 2.0 controller
- Out-of-Box WOL support
- PXE 2.0 Remote Boot support
- 484 PBGA Package

Typical Application

The following figure shows a typical BCM5714S-based LOM design.

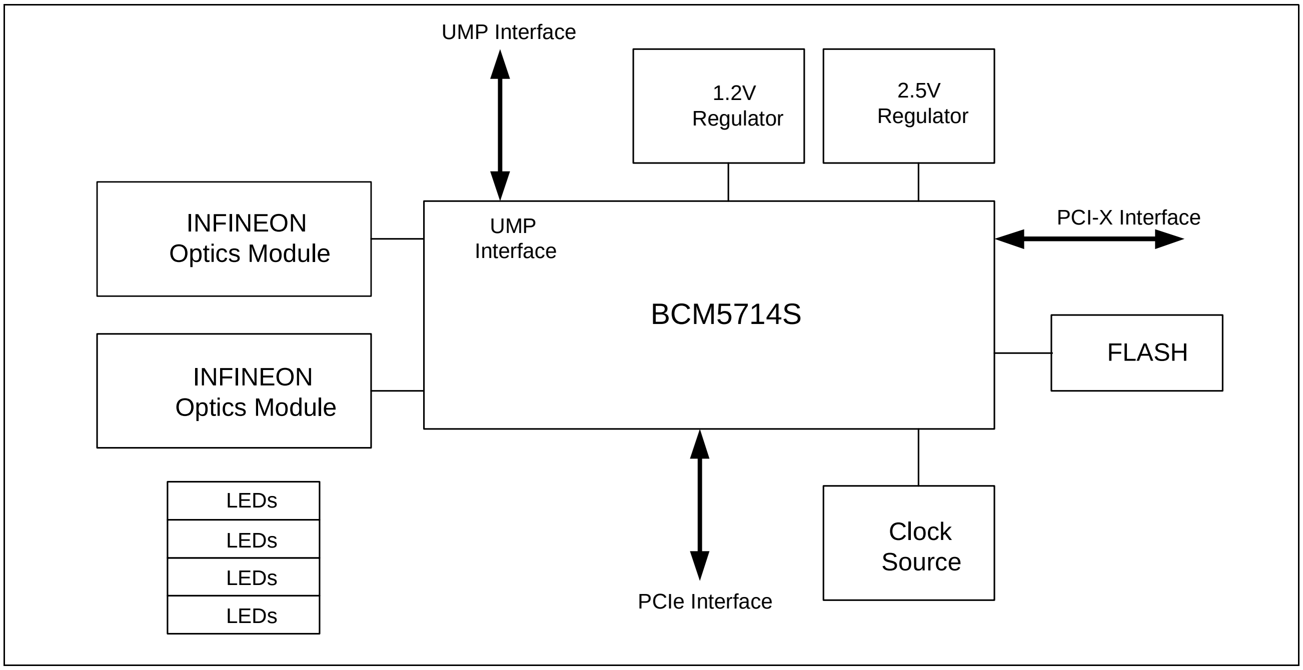


Figure 16: Typical BCM5714S-Based LOM Design Block Diagram

Table 19: BCM5714S NIC Part Component Breakdown

Part Component	***Description***
Optics module	Multimode or single mode fiber optic physical layer.
Clock source	A crystal oscillator generates a 25-MHz clocking signal.
Voltage regulators	Internal voltage regulator controllers provide control for external PNP controlled supplies for 2.5V and 1.2V.
MAC	The BCM5714S MAC.
EEPROM/Flash	The non-volatile storage for firmware boot code, MAC address, VPD, and PXE code. The NVRAM is read after the MAC is reset.

BCM5715C DUAL-MAC CHIP WITH INTEGRATED TRANSCEIVER

The BCM5715C is a fourth-generation 10/100/1000BASE-T dual-port LAN controller for high performance server applications. The device combines dual triple-speed IEEE 802.3 compliant MACs with dual 10/100/1000 Ethernet Transceivers (PHY), a PCIe to PCI-X Internal bridge for connecting to the two MACs, a UMP, and on-chip Frame Buffer Memory in a single device. The device is fabricated in a 1.2V CMOS process providing a low-power system solution.

The BCM5715C provides a x4 PCIe host interface and supports one PCIe to PCI-X internal bridge function, and two Gigabit Ethernet controller functions. Each function has its own PCI configuration space. Each MAC function has a dedicated on-chip high-performance RISC processor for custom frame processing. The BCM5715C also supports a UMP interface for high-speed system management traffic. Following are the important features of BCM5715C device.

- Dual 10BASE-T/100BASE-TX/1000BASE-T full/half-duplex MACs
- Dual 10BASE-T/100BASE-TX/1000BASE-T Ethernet PHY Transceivers
- x4 PCIe host interface with x4 PCIe to PCI-X internal bridge for connecting to dual MACs and PCI-X internal to PCI-X v1.0 bridge.
- UMP
- TCP segmentation, IP fragmentation and reassembly
- IEEE 802.1Q VLAN Tag Support
- IEEE 802.1p Layer2 priority encoding
- IEEE 802.3x Flow Control
- CPU offload functions: TCP, IP, UDP checksum
- 32 KB dedicated receive buffer
- 22 KB transmit buffer to store TX packets and IPMI packets
- Jumbo frame support (9 KB)
- ASF 2.0 support
- IPMI Pass-through support
- ACPI 1.1 compliant
- 36 KB of scratch pad memory
- One Send Ring, One Receive Return Ring, and One Standard Receive Producer Ring
- SMBus 2.0 controller
- Out-of-Box Wake-On LAN Support
- PXE 2.0 Remote Boot Support
- 484 PBGA Package

TYPICAL APPLICATION

The following figure shows a typical BCM5715C-based LOM design.

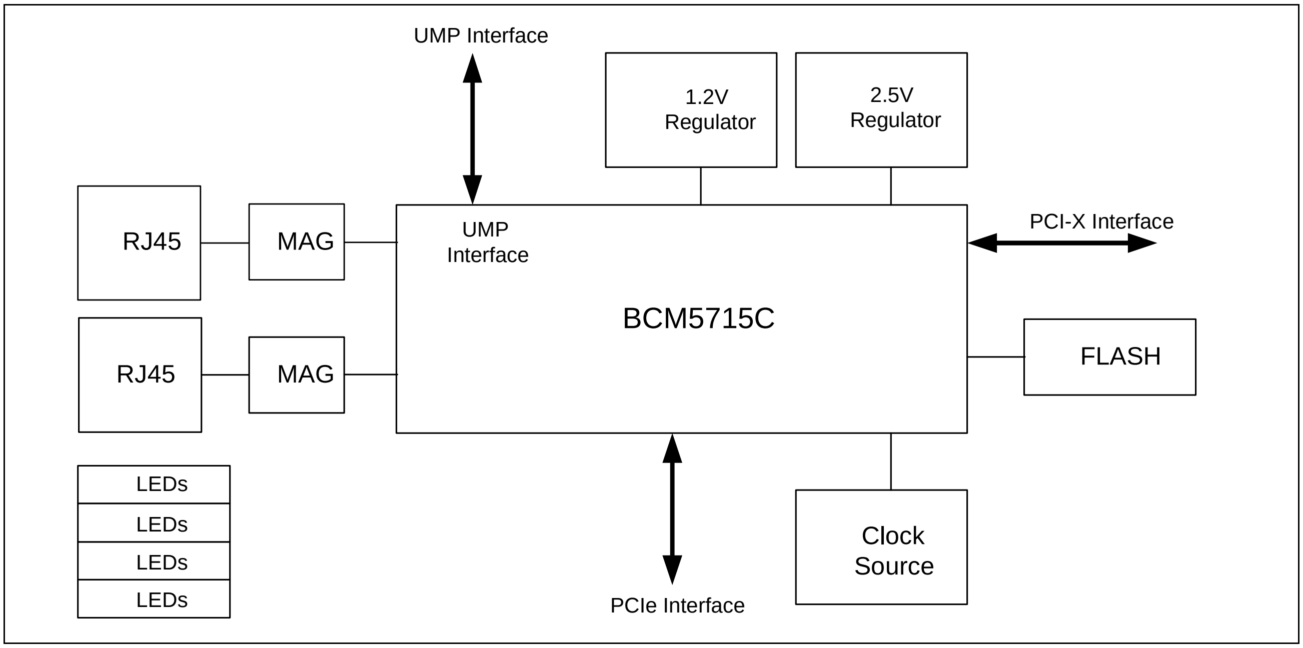


Figure 17: Typical BCM5715C-Based LOM Design Block Diagram

Table 20: BCM5715C NIC Part Component Breakdown

Part Component	*Description*
RJ45	The physical connector for category 5 twisted-pair cabling.
Magnetics	The pulse H5007 isolates the physical layer from voltage events such as sags, swells, and transients. The magnetics module also compensates for impedance mismatches between the cabling and physical layer.
Clock source	A crystal oscillator generates a 25-MHz clocking signal.
Voltage regulators	Internal voltage regulator controllers provide control for external PNP controlled supplies for 2.5V and 1.2V.
MAC	The BCM5715C MAC.
EEPROM/Flash	The non-volatile storage for firmware boot code, MAC address, VPD, and PXE code. The NVRAM is read after the MAC is reset.

BCM5715S DUAL-MAC CHIP WITH INTEGRATED FIBER TRANSCEIVER

The BCM5715S is a fourth-generation 1000BASE-X dual-port LAN controller for high performance server applications. The device combines dual triple-speed IEEE 802.3 compliant MACs with dual SerDes transceivers, a PCIe to PCI-X internal bridge, an UMP and an on-chip frame buffer memory in a single device. The device is fabricated in a 1.2V CMOS process providing a low-power system solution.

The BCM5715S device is same as BCM5715C device except that it has two integrated SerDes transceivers instead of two 10BASE-T/100BASE-Tx/1000BASE-T transceivers. Following are the important features of BCM5715S device.

- Dual 10BASE-T/100BASE-TX/1000BASE-T full/half-duplex MACs
- Dual 1000BASE-X Ethernet SerDes Transceivers
- x4 PCIe host interface with x4 PCIe to PCI-X Internal Bridge for connecting to dual MACs and PCI-X Internal to PCI-X v1.0 Bridge.
- UMP
- TCP segmentation, IP fragmentation and reassembly
- IEEE 802.1Q VLAN Tag Support
- IEEE 802.1p Layer2 priority encoding
- IEEE 802.3x Flow Control
- CPU offload functions: TCP, IP, UDP checksum
- 32 KB dedicated receive buffer
- 22 KB transmit buffer to store TX packets and IPMI packets
- Jumbo frame support (9 KB)
- ASF 2.0 support
- IPMI Pass-through support
- ACPI 1.1 compliant
- 36 KB of scratch pad memory
- One Send Ring, One Receive Return Ring, and One Standard Receive Producer Ring
- SMBus 2.0 controller
- Out-of-Box Wake-On LAN Support
- PXE 2.0 Remote Boot Support
- 484 PBGA Package

Typical Application

The following figure shows a typical BCM5715S-based LOM design.

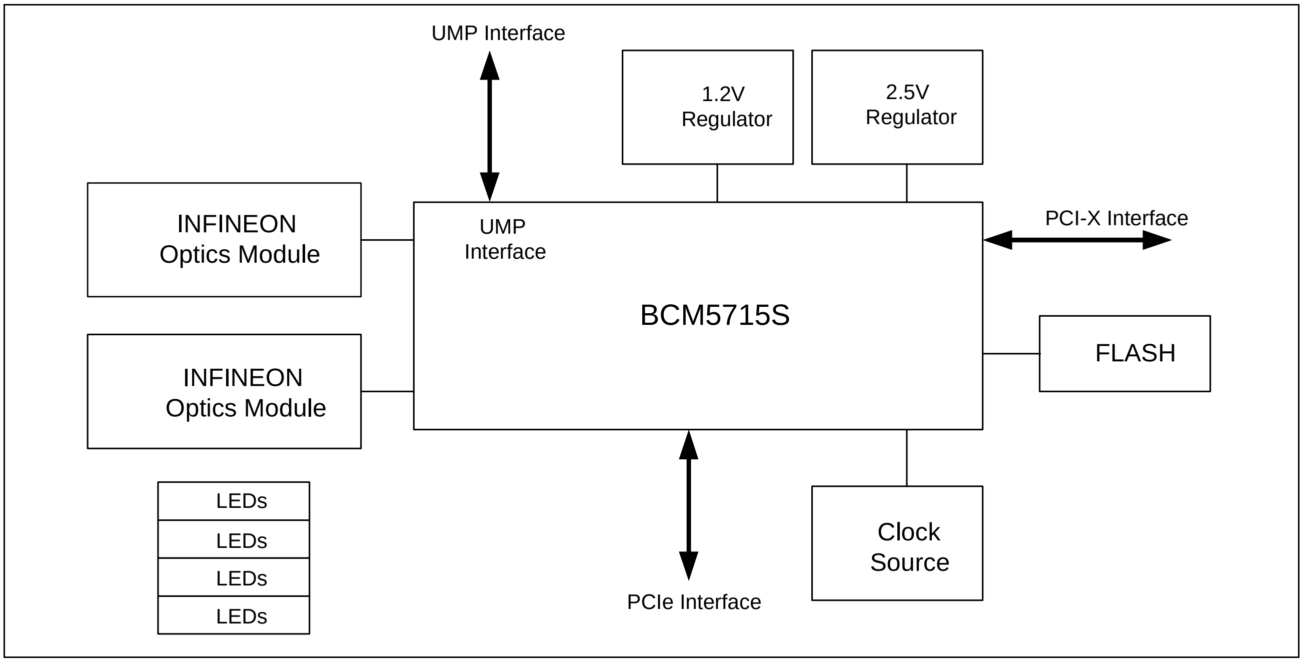


Figure 18: Typical BCM5715S-Based LOM Design Block Diagram

Table 21: BCM5715S NIC Part Component Breakdown

Part Component	***Description***
Optics module	Multimode or single mode fiber optic physical layer.
Clock source	A crystal oscillator generates a 25-MHz clocking signal.
Voltage regulators	Internal voltage regulator controllers provide control for external PNP controlled supplies for 2.5V and 1.2V.
MAC	The BCM5715S dual-port MAC.
EEPROM/Flash	The non-volatile storage for firmware boot code, MAC address, VPD, and PXE code. The NVRAM is read after the MAC is reset.

BCM5752 MAC DEVICE WITH INTEGRATED TRANSCEIVER

The BCM5752 is an eight generation 10/100/1000BASE-T Ethernet LAN controller solution for high-performance network applications. The device combines a triple-speed IEEE 802.3 compliant MAC with a triple-speed Ethernet transceiver, a 1x PCIe bus interface, and on-chip buffer memory in a single device. Following are the important features of BCM5752 device.

- PCIe v1.0a, x1 Link Width Interface
- Single high-speed on-chip RISC Processor
- 10BT/100BTX/1000BT full-duplex and half-duplex MAC
- 10BT/100BTX/1000BT Ethernet PHY Transceiver
- IEEE 802.1Q VLAN Tag Support
- IEEE 802.1p Layer2 priority encoding
- Wake On LAN support meeting ACPI requirements
- TCP Segmentation Support
- IEEE 802.3x Flow Control
- Integrated 64-KB Rx Packet Buffer Memory and 8-KB Tx Packet Buffer Memory
- Programmable Receive Rule Checker
- SMBus Interface supporting ASF 2.0
- Failover and Teaming capabilities
- Serial EEPROM and serial Flash Support
- JTAG Support
- Integrated Trusted Platform Module (TPM) Security Engine Compliant with the Trusted Computing Group Main Specification version 1.2 and Low Pin Count Interface Support.
- 1.2V CMOS with 5V tolerant PCI I/Os
- Available in a 144 fpBGA package (0.8 mm and 1mm pitch)

Typical Application

The following figure shows a typical BCM5752-based NIC board layout.

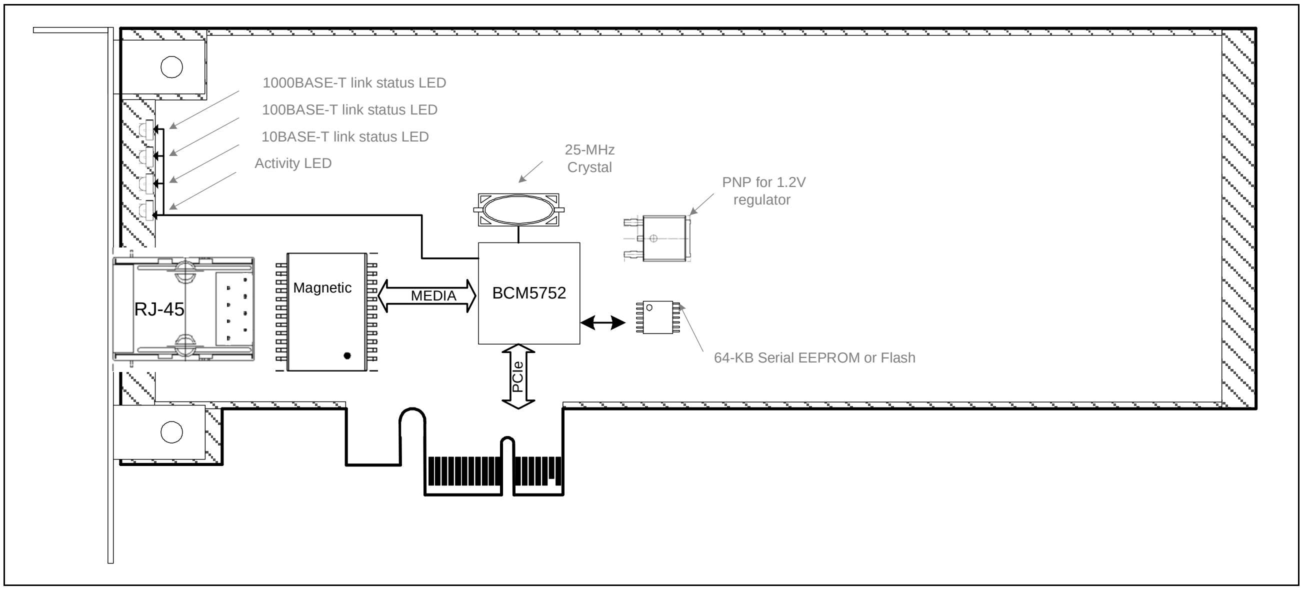


Figure 19: Typical BCM5751-Based NIC Board Block Diagram

Table 22: BCM5751 NIC Part Component Breakdown

Part Component	*Description*
RJ45	The physical connector for category 5 twisted-pair cabling.
Magnetics	The pulse H5007 isolates the physical layer from voltage events such as sags, swells, and transients. The magnetics module also compensates for impedance mismatches between the cabling and physical layer.
Clock source	A crystal oscillator generates a 25-MHz clocking signal.
Voltage regulators	Internal voltage regulator controllers provide control for external PNP controlled supplies for 2.5V and 1.2V.
MAC	The BCM5752 MAC.
PCIe	This specification defines a protocol for bus master controller and target data movement. The MAC is a bus master controller and may move data without CPU intervention.
EEPROM/Flash	The non-volatile storage for firmware boot code, MAC address, VPD, and PXE code. The NVRAM is read after the MAC is reset.

PROGRAMMING ASPECTS

See Table 2 on page 5 for the revision levels of the BCM57XX family. Host software can use the PCI Revision ID and Chip ID information in the PCI configuration registers to determine the revision level of the BCM57XX chip on the board to load the appropriate workarounds described in the errata sheets.

Choice of host access mode determines mailbox priority (see "Configuration Space" on page 178 in Section 9: "PCI"):

- Host standard or flat mode uses the high-priority mailboxes (see "High-Priority Mailboxes" on page 370).
- Indirect mode uses the low-priority mailboxes (see "Low-Priority Mailboxes" on page 490).

See "Device Control" on page 146 for the procedure to initialize this device.

Section 3: Hardware Architecture

THEORY OF OPERATION

The functional block diagram below in Figure 20 shows the major functional blocks and interfaces of the BCM57XX NetXtreme family of gigabit Ethernet MACs. There are two packet flows: MAC-transmit and receive. The device's DMA engine will bus-master packets from host memory to device local storage, and vice-versa. The BCM57XX's host bus interface is compliant with PCI, PCI-X, and PCIe standards (depending on the particular BCM57XX product). The RX MAC moves packets from the integrated PHY into device 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 data from device internal memory to the PHY. Both flows operate independently of each other in full-duplex mode. On-chip RISC processors are provided for running value-added firmware that may be used for custom frame processing. The on-chip RISCs operate independently of all the architectural blocks; essentially, RISCs are available for the auxiliary processing of data streams.

See "Features" on page 11 for typical applications of the chips in the BCM57XX family.

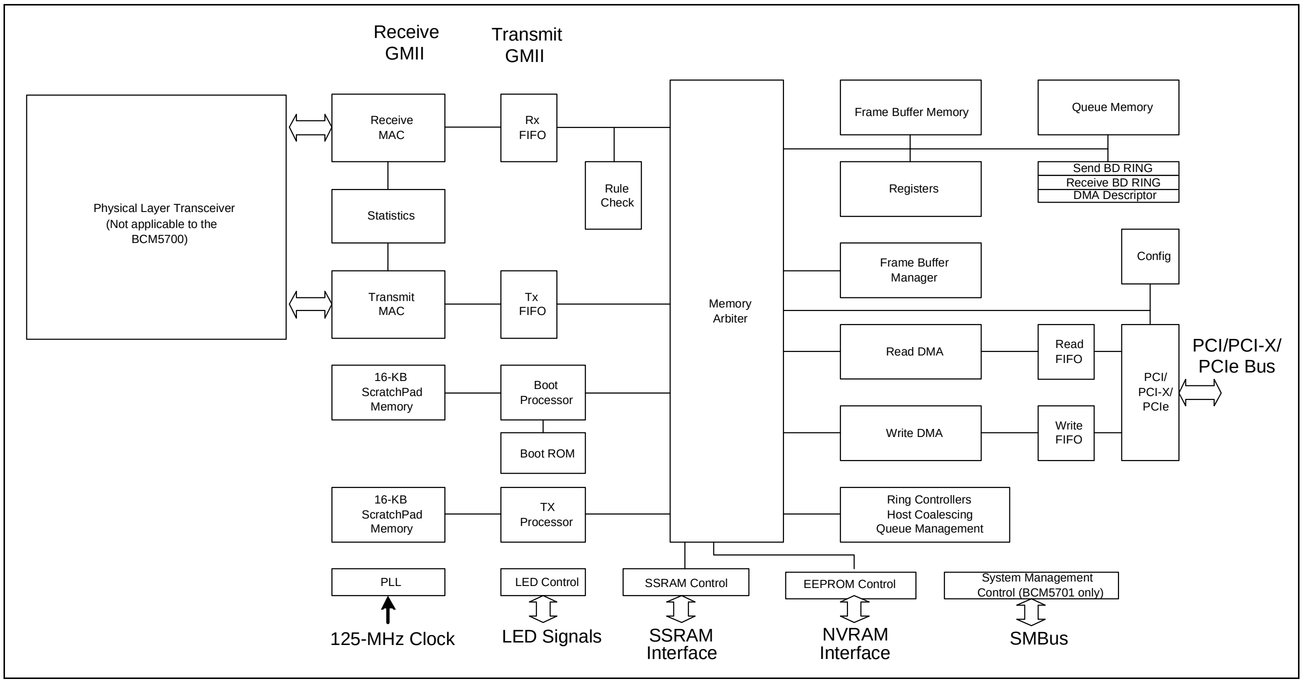


Figure 20: Functional Block Diagram

RECEIVE DATA PATH

RX ENGINE

The receive engine (see the following figure) activates whenever a packet arrives from the PHY.

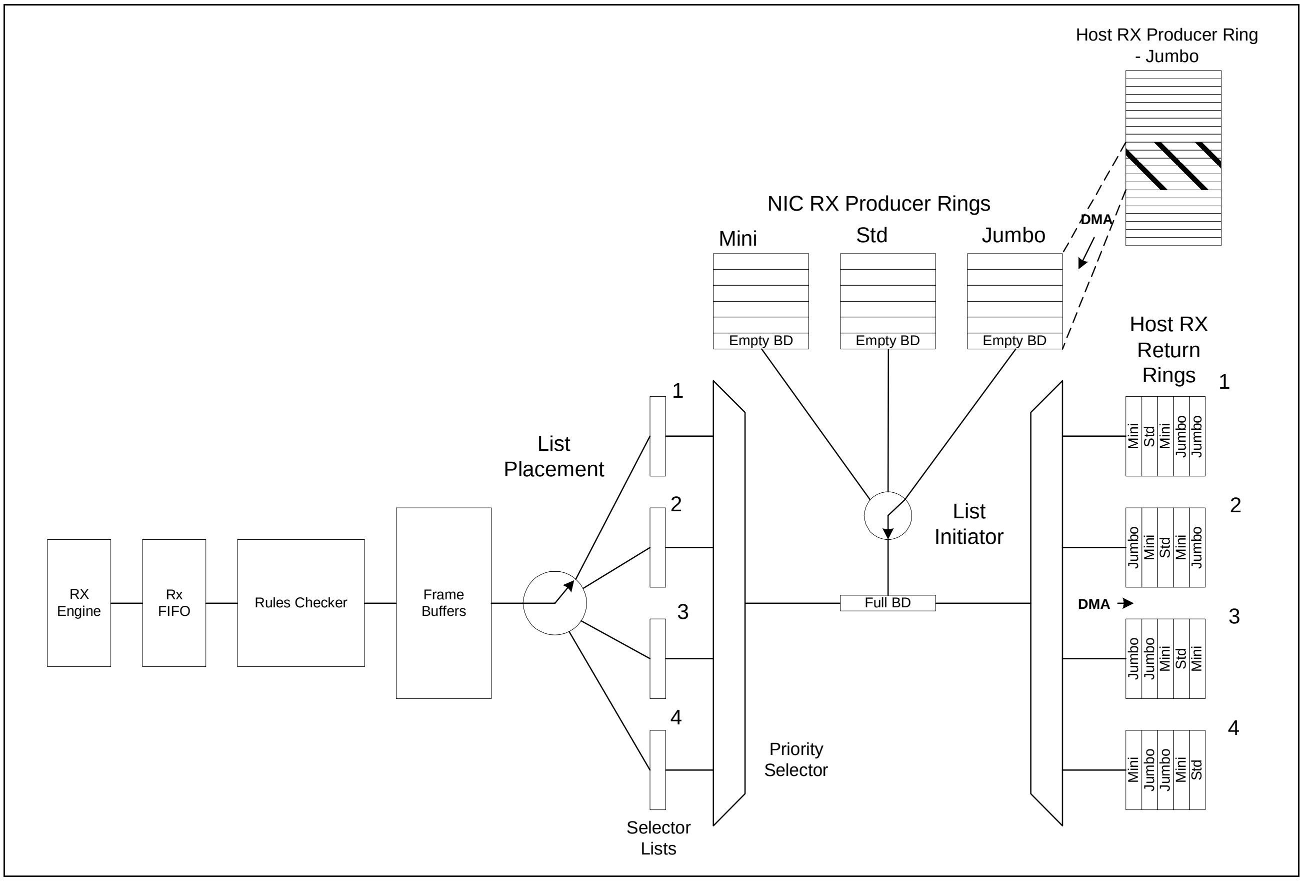


Figure 21: Receive Data Path

The receive engine performs the following four functions:

- Moves the data from the PHY to an internal FIFO
- Moves the data from the FIFO to NIC internal memory
- Classifies the frame and checks it for rules matches
- Performs the offloaded checksum calculations

RX FIFO

The RX FIFO provides elasticity while data is read from PHY Transceiver and written into internal memory. There are no programmable settings for the RX FIFO. This FIFO's operation is completely transparent to host software.

RULES CHECKER

The rules checker will examine frames. Once a frame has been examined, the appropriate classification bits are set in the buffer descriptor. The rules checker is part of the RX data path and the frames are classified during data movement to NIC memory. The following frame positions may be established by the rules checker:

- IP Header Start Pointer
- TCP/UDP Header Start Pointer
- Data Start Pointer

RX LIST PLACEMENT

The RX List Placement function determines one of the 16 receive lists the frame should be placed on. Then, the RX List Placement block adds the frame to the appropriate list. The selection is done based on a class value in the frame descriptor. There are no configuration registers for this block beyond the mode control register (see "Mode Control Register (Offset 0x6800)" on page 502). The BCM5705, BCM5788, BCM5721, BCM5751, BCM5714C, BCM5714S, BCM5715C, BCM5715S, and BCM5752 MACs have only one receive list and hence all the received frames that are not discarded by rules checker are placed into this single receive list.

RX LIST INITIATOR

The RX List Initiator function activates whenever the receive producer index for any of receive Buffer Descriptor (BD) rings is written. This value is located in one of the Receive BD producer mailboxes. The host software writes to the producer mailbox and causes the RX Initiator function to enqueue an internal data structure/request, which initiates the DMA of one or more new BDs to the NIC. The actual DMAs generated depend on the comparison of the value of the received BD host producer index mailbox, the NIC's copy of the received BD consumer index, and the local copy of the received BD producer index. The RX List Initiator will select the BD from either mini or standard or jumbo ring based on frame size. Note that mini ring is supported only in BCM5700 MAC device. Also note that the jumbo ring is not supported in BCM5705, BCM5788, BCM5721, BCM5751, BCM5714C, BCM5714S, BCM5715C, BCM5715S, and BCM5752 MACs

TRANSMIT DATA PATH

TX MAC

The Read DMA engine moves packets from host memory and then into internal NIC memory (see Figure 22). Once the entire packet is available, the transmit MAC will be activated.

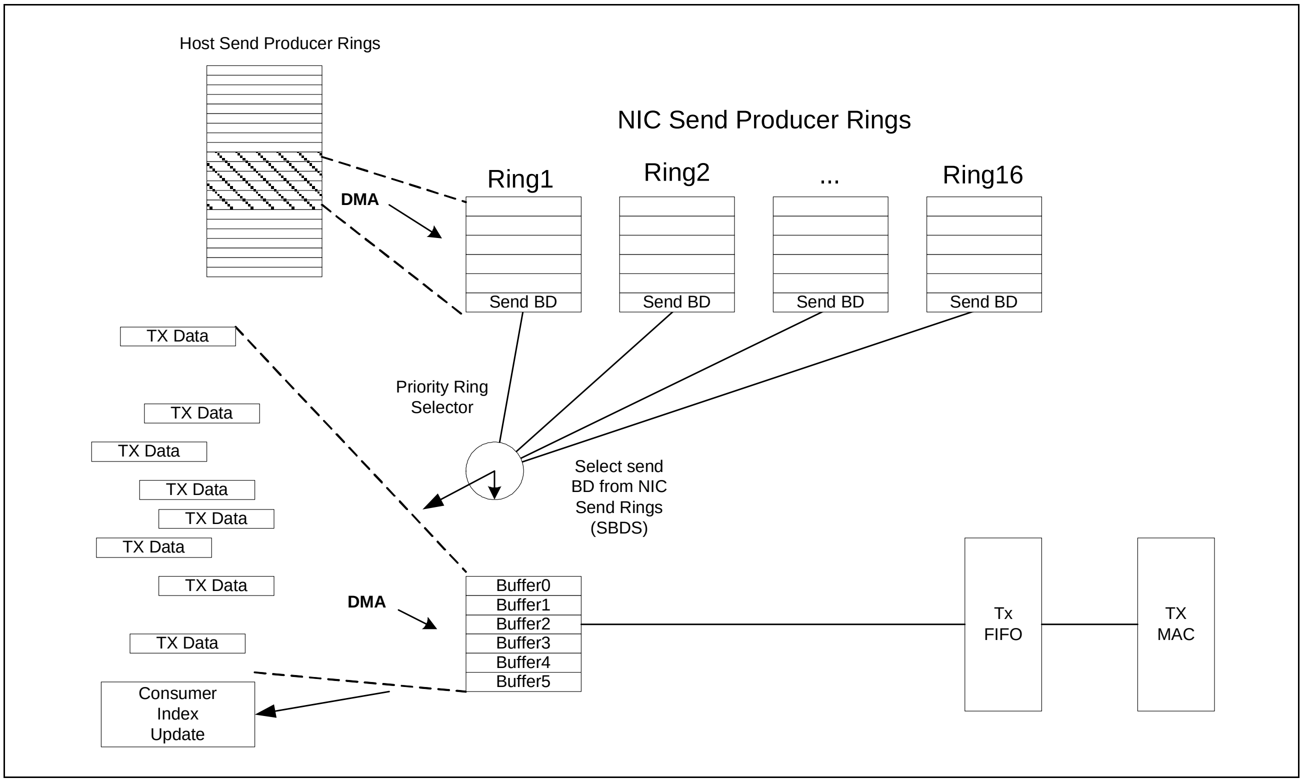


Figure 22: Transmit Data Path

The transmit MAC is responsible for the following functions:

- Moving data from NIC internal memory into TX FIFO
- Moving data from TX FIFO to PHY
- Checksum substitutions (not calculation)
- Updating statistics

TX FIFO

The TX FIFO provides elasticity, while data is moved from device internal memory to PHY. There are no programmable settings for the TX FIFO. This FIFO's operation is completely transparent to host software.

TX Priority Ring Selector

The TX Priority Ring Selector chooses a send BD ring in accordance with a priority level. For the 16 send rings, ring 1 has the highest priority and ring 16 has the lowest priority. All buffer descriptors required to complete a single frame are consumed. At no time are portions of frames intermixed—only the buffer descriptor(s) associated with one frame are consumed. The BCM5705, BCM5788, BCM5721, BCM5751, BCM5714C, BCM5714S, BCM5715C, BCM5715S, and BCM5752 MACs have only one send ring.

DMA Read

Read Engine

The DMA read engine (see the following figure) activates whenever a host read is initiated by the send or receive data paths.

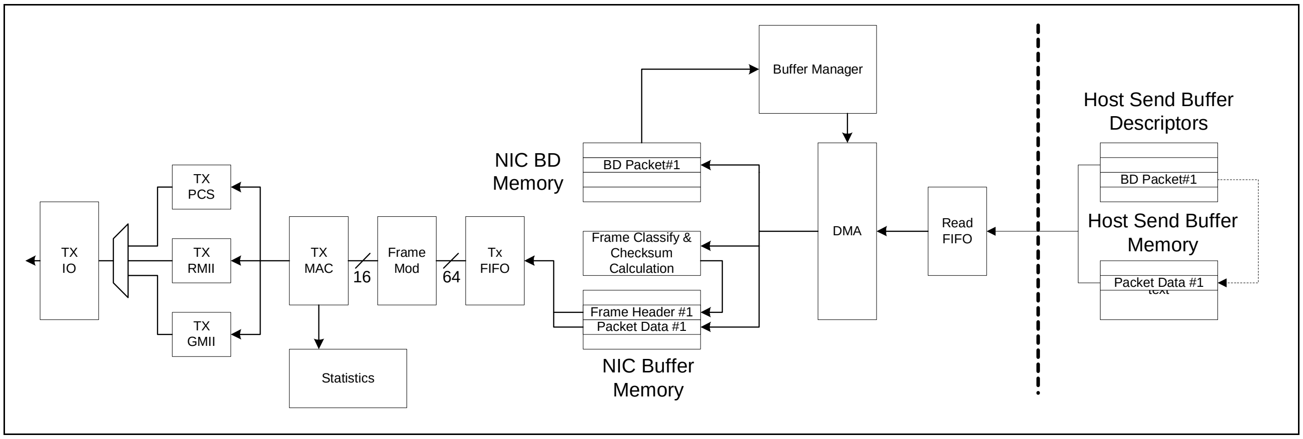


Figure 23: DMA Read Engine

The DMA read engine dequeues an internal data structure/request and performs the following functions:

- DMAs the data from the host memory to an internal Read DMA FIFO
- Moves the data from the Read DMA FIFO to NIC internal memory
- Classifies the frame
- Performs checksum calculations
- Copies the VLAN tag field from the DMA descriptor to the frame header

READ FIFO

The read FIFO provides elasticity during data movement from host memory to device local memory. The BCM57XX memory arbiter is a gatekeeper for multiple internal blocks; several portions of the architecture may simultaneously request internal memory. The PCI Read FIFO provides a small buffer for the data read from host memory while the Read DMA engine requests internal memory via the memory arbiter. The data is moved out of the read DMA FIFO into device local memory once a memory data path is available. The FIFO isolates the PCI clock domain from the device clock domain. This reduces latency internally and externally on the PCI bus. The PCI Read DMA FIFO holds 512 bytes. The operation of the read DMA FIFO is transparent to host software.

BUFFER MANAGER

The buffer manager maintains pools of internal memory used by transmit and receive engines. The buffer manager has logic blocks for allocation, free, control, and initialization of internal memory pools. The DMA read engine requests internal memory for BDs and frame data. Figure 23 on page 53 shows the transmit data path using the DMA Read Engine. The read DMA engine will also fetch Rx BDs for the receive data path.

DMA WRITE

WRITE ENGINE

The DMA write engine (see the following figure) activates whenever a host write is initiated by the send or receive data paths.

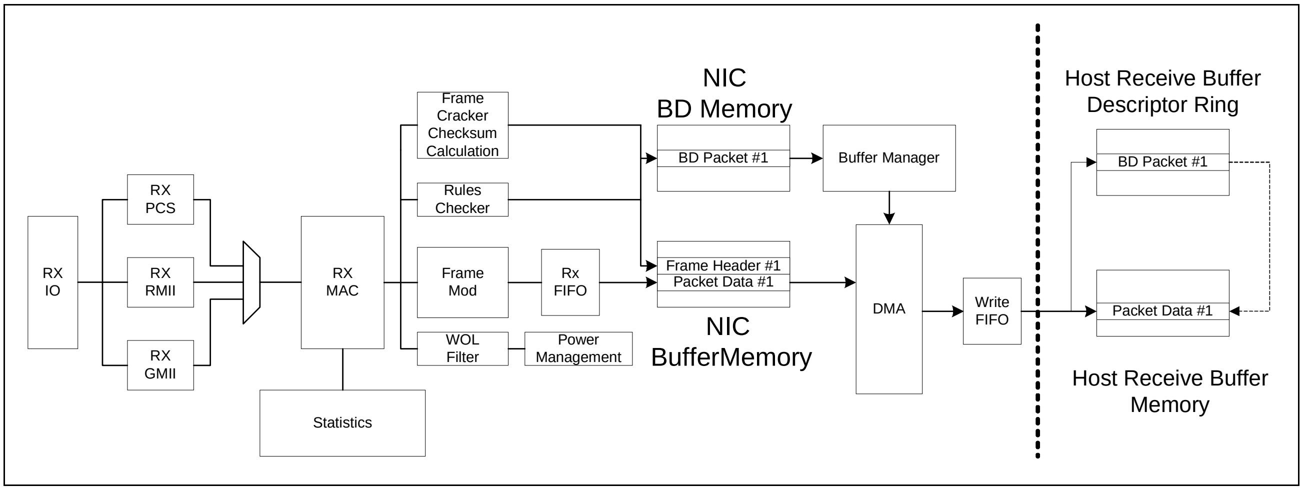


Figure 24: DMA Write Engine

The DMA write engine dequeues an internal request and performs the following functions:

- Gathers the data from device internal memory into the write DMA FIFO
- DMAs the data to the host memory from the write FIFO
- Performs byte and word swapping
- Interrupts the host using a line or message signaled interrupt

WRITE FIFO

The write FIFO provides elasticity during data movement from device memory to the host memory. The write FIFO absorbs small delays created by PCI/PCI-X/PCIe bus arbitration. Multiple devices may be granted PCI bus cycles during a write DMA operation. The BCM57XX family uses the write FIFO to buffer data, so internal memory arbitration is efficient. Additionally, the FIFO isolates the PCI clock domain from the device's clock domain. This reduces latency on the PCI bus during the write operation (wait states are not inserted while data is fetched from internal memory). The PCI write DMA FIFO holds 512 bytes. The operation of the write DMA FIFO is transparent to host software.

BUFFER MANAGER

The buffer manager maintains pools of internal memory used in transmit and receive functions. The buffer manager has logic blocks for allocation, free, control, and initialization of internal memory pools. The receive MAC is given higher priority over the transmit MAC internally; it is very costly to drop received packets. Whereas, the transmit MAC may be stalled temporarily until internal buffers become available. The receive MAC will request NIC memory so inbound frames can be buffered. The DMA write engine requests a small amount of internal memory for DMA and interrupt operations. The usage of this internal memory is transparent to host software, and does not affect device/system performance.

SYSTEM MANAGEMENT BUS (NOT APPLICABLE TO BCM5700)

Figure 25 shows the architecture of the SMBus and the location of the ASF in the system.

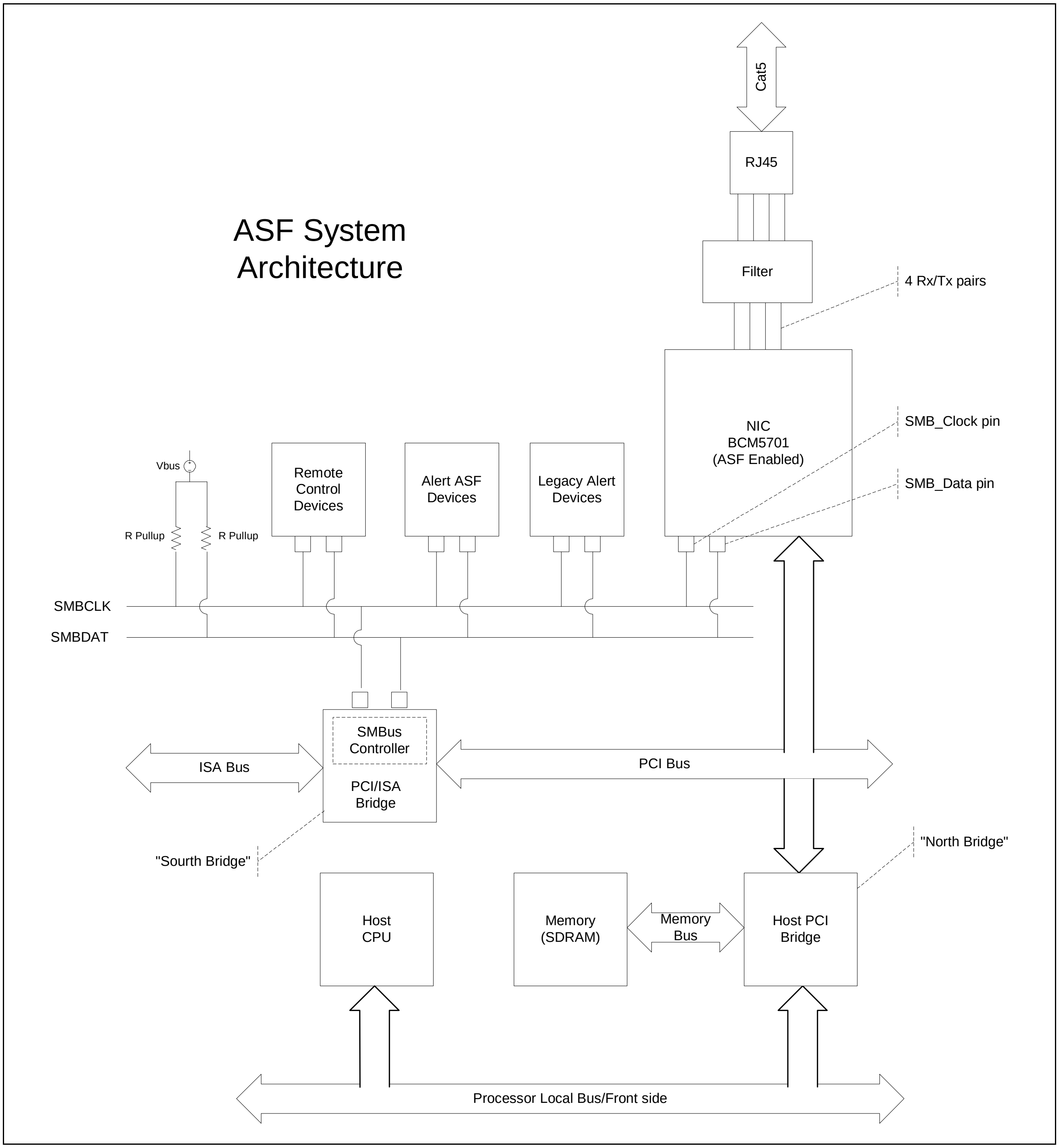


Figure 25: ASF System Architecture

OVERVIEW

The System Management block of the BCM57XX MAC Transceivers contains all of the hardware necessary to support a primarily firmware implementation of the ASF protocol. This hardware includes six dedicated timers, a low-level SMBus 2.0 compliant interface, registers for driving the SMBus interface, and a global control and event register. The system management block can also be used to support additional (non-ASF) management technologies such as IPMI.

TIMERS

The System Management block contains five countdown timers that stop when they reach zero and one free-running timestamp counter. The countdown timers all have a corresponding event bit in the control and event register. The event bit is set when the counters transition from a value of one to a value of zero. The timestamp counter starts when its enable bit is set in the control register and rolls over when it reaches its maximum value.

SMBUS INTERFACE

The SMBus interface provides a serial interface for byte-wide words and allows the firmware to directly control the operation of the bus. The SMB Enable bit (bit 12) in the ASF Control register (see "ASF Control Register (Offset 0x6C00)" on page 522) enables the interface.

The interface is capable of master and slave modes of operation. A master operation is initiated by writing the first byte of data to the SMB Output register (see "SMBus Output Register (Offset 0x6C08)" on page 524), along with a START bit, a RDY bit, and any other desired control bits for that transaction. Subsequent data bytes to be transmitted are then written to the same register with the RDY bit set. For the last transmit byte, the LAST bit and RDY bit are set. The transmit bytes are transferred from the SMB Output register to an internal FIFO whenever there is space, and the RDY bit is cleared when this transfer takes place. Therefore, the RDY bit is used to determine if there is space to write the next transmit byte. The START bit is cleared by the hardware when the transmit portion of the transfer is finished. At the same time, the status of the transmit portion of the transfer is provided in the STATUS field.

The transmit portion of the transfer could end at any time due to arbitration loss, no-ack from the target, or assorted error conditions. If the transfer is a read transaction, then the read data can now be retrieved from the SMB Input register. The data field of this register is valid when the RDY bit is set. The input portion of the SMBus interface also has an internal FIFO, and data is transferred from this FIFO to the SMB Input register (see "SMBus Input Register (Offset 0x6C04)" on page 524) automatically whenever the RDY bit is clear. If the SMB AUTOREAD bit (bit 15) of the ASF Control register is set, then the RDY bit is cleared whenever the SMB Input register is read, thus allowing the next data byte to be loaded. The SMB Input register has a DONE bit that is set when the receive portion of the transfer is completed. A status field is also provided when the DONE bit is set. The DONE bit must be cleared by firmware after the status is retrieved. Also, the firmware must flush all of the data out of the FIFO.

A slave mode operation starts whenever the SMBus interface detects activity on the bus, or if a master operation loses arbitration during the first byte of the transfer. The interface does address filtering on the incoming data by looking at the first byte of the transfer (if the SMB ADDR Filter bit is set in the ASF Control register). The address is compared to two different fields in the ASF Control register, and to the value 0x00 if the SMB Enable ADDR Zero bit is set. If an address match is found, then the interface acknowledges the incoming byte, and transfers the byte to the input FIFO. The RDY and DONE bits in the SMB Input register behave in this mode basically the same as with master reads. If the slave access calls for read data, then the firmware would send data to the interface via the SMB Output register in a manner similar to master-write accesses.

A bit-bang mode for this interface is also provided. To enable this mode, set the SMB Bit Bang Enable bit in the ASF Control register and clear the SMB Enable bit. Control of the bus is accomplished by directly manipulating the SMB Bit Bang bits in the SMB Output register as defined in the register definition.

SMBus Connector

The SMBus signals were added to the PCI physical connector with a PCI v2.2 Engineering Change Notification (ECN) titled *Addition of the SMBus to the PCI Connector*. Prior to this ECN, network card vendors used a two-pin header and twisted management cable. This solution complicated installation and increased manufacturing costs. Two pins previously marked reserved on the PCI connector side-A have now been allocated for SMBus signaling. Pin #40 on PCI Side-A is now defined as the SMBCLK signal. Pin #41 on PCI Side-A is now defined as the SMBDATA signal. These SMBus signals are defined in both a 3.3V and 5.0V PCI signaling application. The SMBus signals are exempt from a few requirements regarding loading and pull-ups, and the programmer is encouraged to read the ECN on the PCI SIIG website. The BCM57XX MAC-Transceiver's SMBus interface meets the high power requirements stated in the SMBus 2.2 specification.

SMBus Data Link

The BCM57XX MAC Transceiver's SMBus interface is compliant with the SMBus 2.0 specification. Each transaction contains a start and stop delimiter. The SMBus interface is very similar to I^{2}C with a few differences. For example, SMBus masters may enumerate the bus and begin a device address resolution; should two devices have a conflicting address. A start delimiter is detected when the SMBCLK is sampled high and the SMBDATA signal transitions from high to low. A stop delimiter is detected when the SMBCLK is sampled high and the SMBDATA signal transitions from low to high. See Figure 26.

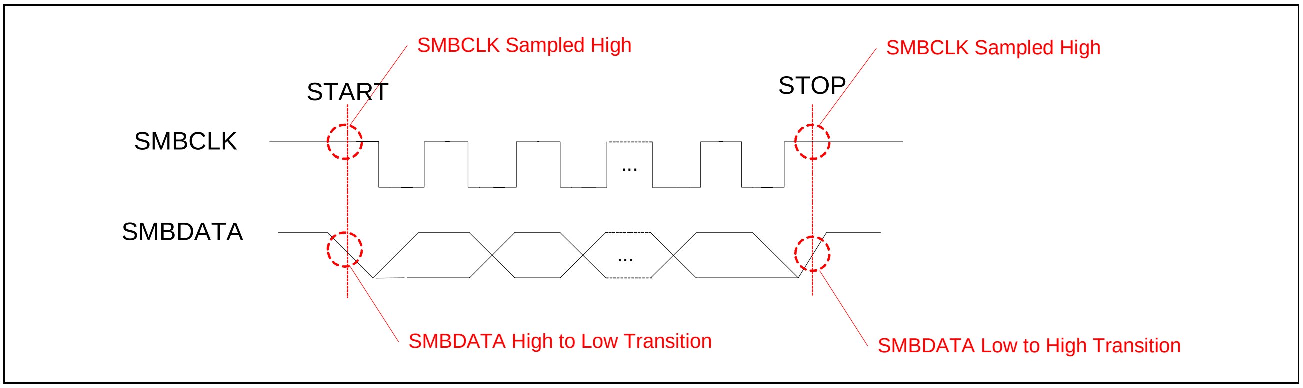


Figure 26: SMBus Start and Stop Conditions

A SMBus transactions consist of several phases. A SMBus master will start a transaction by driving SMBDATA low and keeping SMBCLK high—a start condition. If two masters are driving the SMBus simultaneously, the masters must arbitrate for the bus. The programmer is encouraged to read section 4.3.2 in the SMBus 2.0 specification. The master that drives SMBDATA high, yet detects SMBDATA driven low (contention condition) by a second master, loses bus arbitration. The arbitration may continue into the address/data phases and past the start condition. Both masters may drive the bus low simultaneously for an indeterminate number of clocks until one master senses bus contention. See Figure 27 on page 59.

Next, a master must transmit the slave address—the target devices, which decode the read/write transaction. There are reserved slave addresses that should not be used by any SMBus devices. The programmer is encouraged to look at table 4 in section 5.2 of the SMBus 2.0 specification. After the slave address is driven on the bus, the master will indicate whether the transaction is a read or write. The slave device drives an ACK or NAK on the SMBDATA line for the transaction phase. The master will sample the ACK or NAK accordingly. If the master does sample a NAK, then a Stop condition must be generated. The master must stop/terminate the transaction. See Figure 28 on page 59.

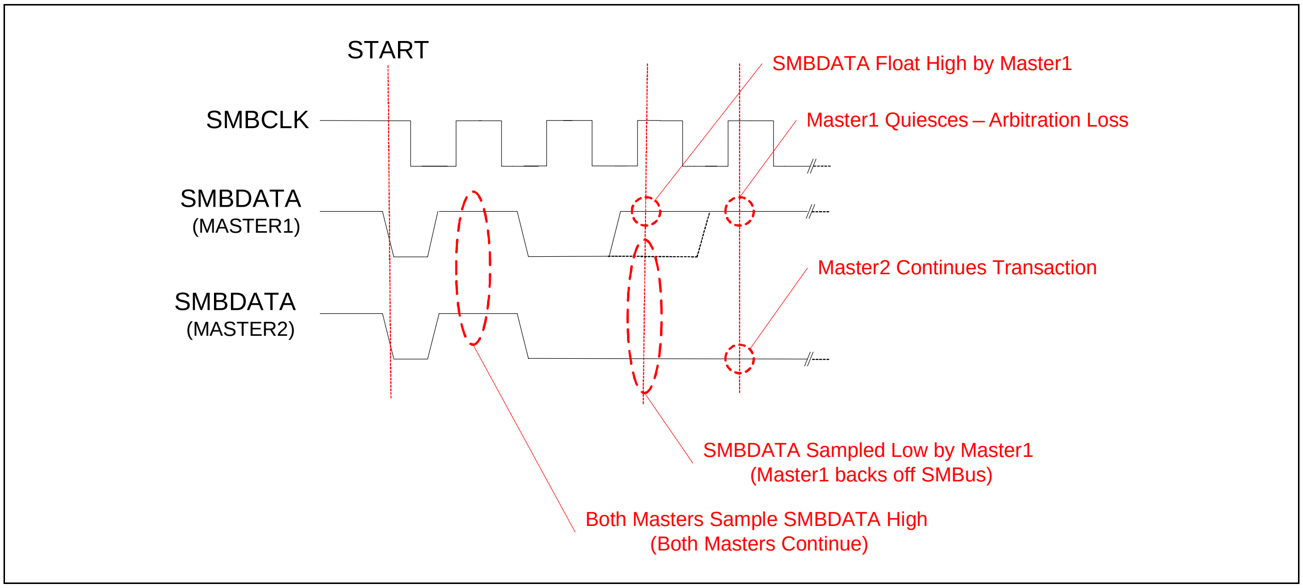

START
SMBCLK
SMBDATA
(MASTER1)
SMBDATA
(MASTER2)
SMBDATA Float High by Master1
Master1 Quiesces – Arbitration Loss
Master2 Continues Transaction
SMBDATA Sampled Low by Master1
(Master1 backs off SMBus)
Both Masters Sample SMBDATA High
(Both Masters Continue)

Figure 27: Two Masters Arbitrate for SMBus

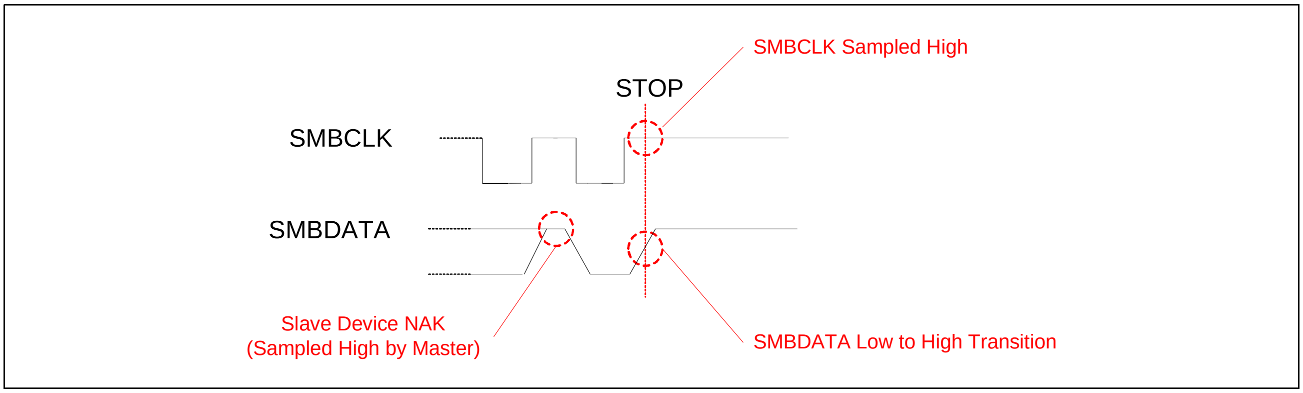


Figure 28: Master Stops Transaction After Slave NAKs

Depending on the type of transaction, the master sources the data byte on the SMBDATA lines:

- Write transactions: the master sources the data byte and the slave sinks/latches the data. The slave must ACK/NAK write transactions.
- Read transactions: the slave sources the data byte and the master sinks/latches the data. The master must ACK/NAK read transactions. Also, the master must drive the stop delimiter, which releases the SMBus and terminates the transaction.

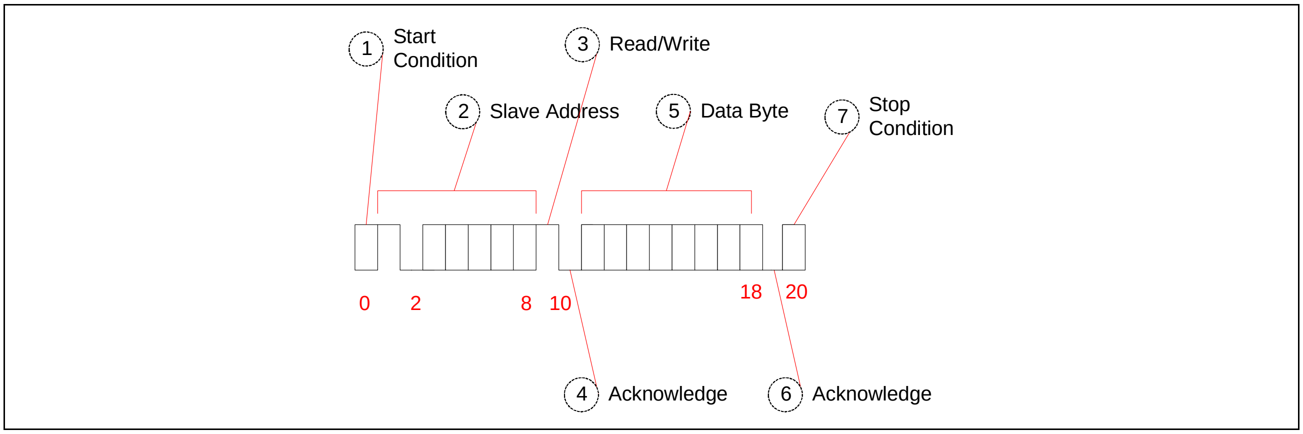


Figure 29: SMBus Transaction Phases

SMBus Clock

The BCM57XX SMBus interface supports clock stretching. Slave devices may clock low to extend the SMB_CLK signal, and the BCM57XX family absorbs the stretch latency. The programmer should refer to section 4.3.3 of the SMBus 2.0 manual for further details on this technology. The BCM57XX will maintain a minimum compliant frequency of 10 kHz when clock is low extending.

The BCM57XX family's clock period (master mode) is 11 µs—a 90.9 kHz frequency. The SMB_CLK signal is driven low T_{Low} for 5 µs. The BCM57XX family floats the SMB_CLK high T_{High} for a maximum of 6 µs before the SMB_CLK is driven low again. The low and high times are not guaranteed symmetric, since the rise may vary due to external pull-up resistors or current sources. The rise time T_R on SMB_CLK will vary from 300 ns to 1 µs. The fall time T_F is roughly 500 ns, but also may vary. See the following figure.

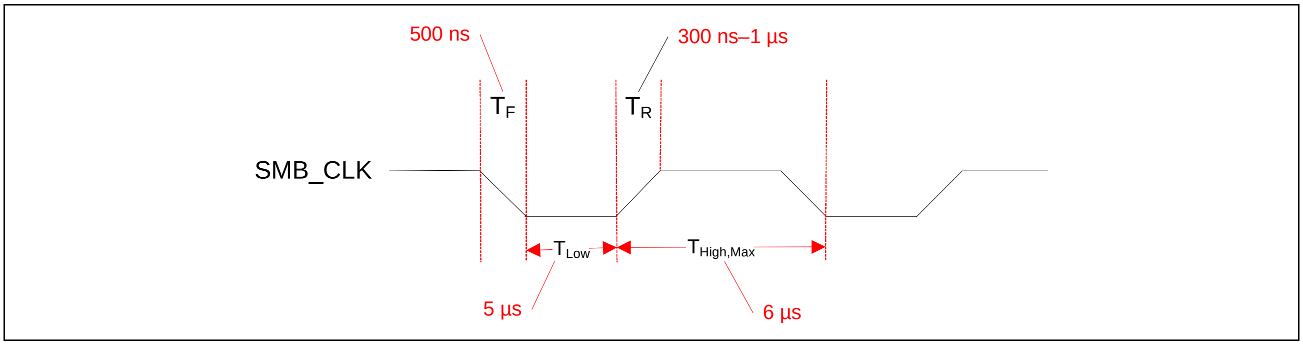


Figure 30: SMB_CLOCK Period (Master Mode)

Events

The SMB Attention bit and the five timeout bits in the ASF Control register are all ORed together to form a single ASF_ATTN signal. Depending on the value of the ASF Attention Location field, this signal may then be mapped into one of several bit positions in the RX CPU Event or TX CPU Event registers.

Other ASF Actions

Most of the ASF required actions other than those described above are to be handled by the firmware in the chips internal RISCs. One exception to this would be detection of the incoming RMCP packets, which would be handled by programming the appropriate rules into the MAC receive rule checker.

LED Control

The BCM57XX family supports four LEDs—one for data traffic in either direction and three for 10/100/1000 Mbps links established. The traffic LED blinks during transmit and receive data movement through the device. The blink rate is programmable with a default of approximately 15 Hz. The link LEDs turn either on or off depending on the link established. All LEDs can be controlled directly by software via the override bits in LED_Control register (see "LED Control Register (Offset 0x40C)" on page 382). When the override bit is off, the link LEDs are automatically set by the hardware as long as link is up:

- 10 Mbps LED—port mode is set to MII and Mode 10 Mbps in MI Status register is set.
- 100 Mbps LED—port mode is set to MII and Mode 10 Mbps in MI Status register is not set.
- 1000 Mbps LED—port mode is set to GMII or TBI.

The link status information is obtained either from auto-polling the PHY if bit 4 (Port Polling) of MI Mode register (see "MI Mode Register (Offset 0x454)" on page 389) is set, or from the LNKRDY input pin if this bit is not set. The polarity of the LNKRDY signal can be toggled by setting bit 10 (Link Polarity) in the Ethernet_MAC_Mode register (see "Ethernet MAC Mode Register (Offset 0x400)" on page 377).

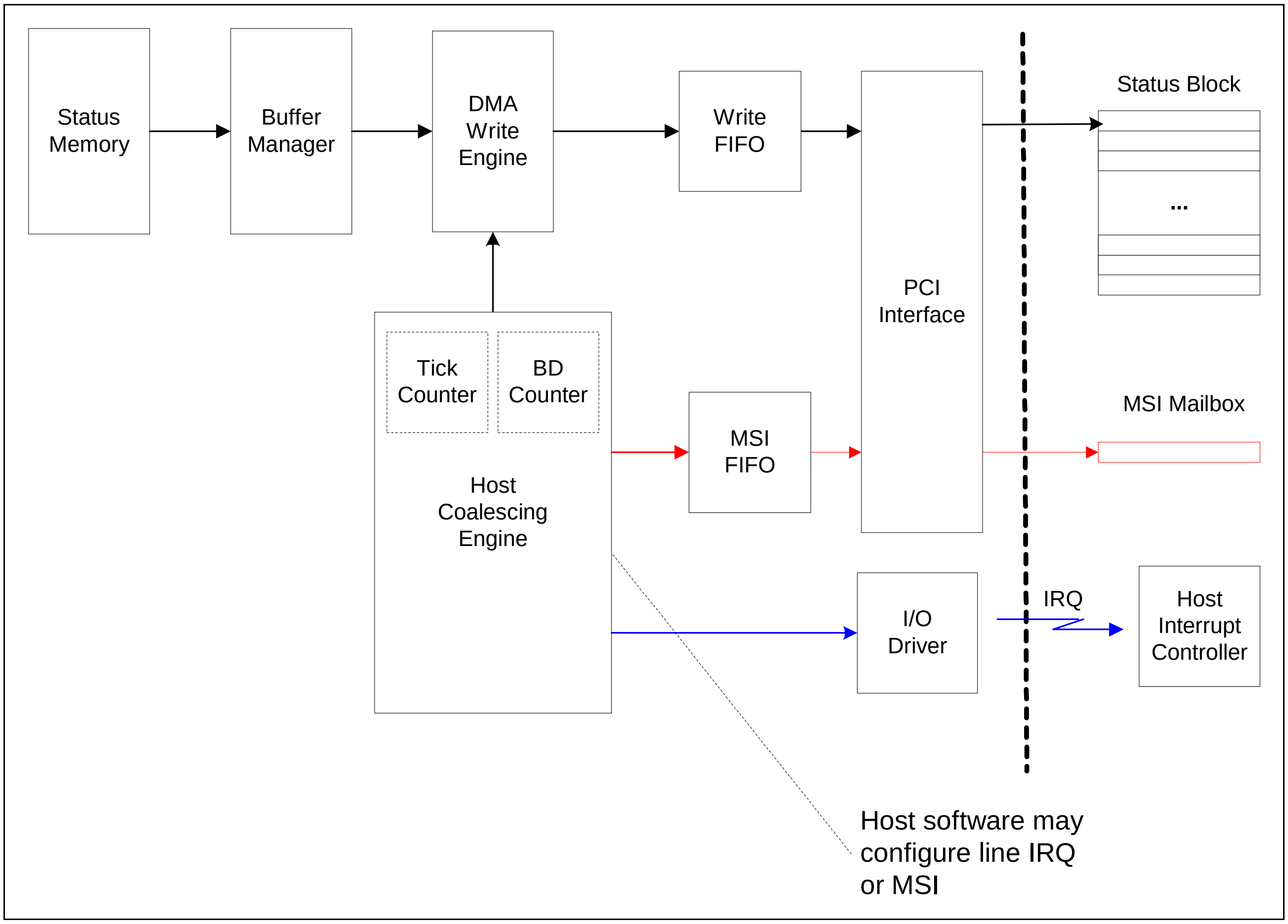
Memory Arbiter

The Memory Arbiter (MA) is a gatekeeper for internal memory access. The MA is responsible for decoding the internal memory addresses that correspond to BCM57XX data structures and control maps. The MA is responsible for accessing both internal and external SSRAM. Should a functional block fault or trap during access to internal/external memory, the MA will handle the failing condition and report the error in a status register. In addition to architectural blocks, the MA provides a gateway for the RISC processors to access local memory. Each RISC has a MA interface that pipelines up to three access requests. The MA negotiates local memory access, so all portions of the architecture are provided with fair access to memory resources. The MA prevents starvation and bounds access latency. Host software may enable/disable/reset the MA, and there are no tunable parameters.

HOST COALESCING

HOST COALESCING ENGINE

The Host Coalescing Engine is responsible for pacing the rate at which the NIC updates the send and receive ring indices located in host memory space. The completion of a NIC update is reflected through an interrupt on the BCM57XX $\overline{\text{INTA}}$ pin or a Message Signalled Interrupt (MSI). Although update criteria are calculated separately, all updates occur at once. This is because all of the ring indices are in one status block, and any host update updates all ring indices simultaneously. The Host Coalescing Engine will trigger based on a tick and/or a frame counter.



Host software may configure line IRQ or MSI

Figure 31: Host Coalescing Engine

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

- The number of BDs consumed for frames received, without updating receive indices on the host, is equal to or has exceeded the threshold set in the Receive_Max_Coalesced_BD register (see "Receive Max Coalesced BD Count (Offset 0x3C10)" on page 454).
- The number of BDs consumed for transmitting frames, without updating the send indices, on the host is equal to or has exceeded the threshold set in the Send_Max_Coalesced_BD register (see "Send Max Coalesced BD Count (Offset 0x3C14)" on page 454). Updates can occur when the number of BDs (not frames) meets the thresholds set in the various coalescing registers (see Section 11: "Interrupt Processing" on page 287 for more information).
- The receive coalescing timer has expired, and new frames have been received on any of the receive rings, and a host update has not occurred. The receive coalescing timer is then reset to the value in the Receive_Coalescing_Ticks register (see "Receive Coalescing Ticks Registers (Offset 0x3C08)" on page 453).
- The send coalescing timer has expired, and new frames have been consumed from any send ring, and a host update has not occurred. The send coalescing timer is then reset to the value.

MSI FIFO

This FIFO is eight entries deep and four bits wide. This FIFO is used to send MSIs via the PCI interface. The host coalescing engine uses this FIFO to enqueue requests for the generation of MSI. There are no configurable options for this FIFO and this FIFOs operation is completely transparent to host software.

STATUS BLOCK

This data structure contains consumer and producer indices/values. Host software reads this control block, to assess hardware updates in the send and receive rings. Two copies of the status block exist. The local copy is DMAed to host memory by the DMA write engine. Host software does not want to generate PCI transactions to read ring status; rather quicker memory bus transactions are desired. The host coalescing engine will enqueue a request to the DMA write engine, so host software gets a refreshed copy of status. The status block is refreshed before a line IRQ or MSI is generated. See "Status Block Format" on page 104 for a complete discussion of the status block.

10/100/1000BASE-T Transceiver (Not Applicable to BCM5700)

Encoder

In 10BASE-T mode, Manchester encoding is performed on the data stream and transmitted on one pair in the twisted pair cable. The multimode transmit DAC performs pre-equalization for 100 meters of CAT 3 cable. In 100BASE-TX mode, the BCM57XX MAC-Transceivers transmit a continuous data stream on one pair in the twisted-pair cable, and receive a continuous data stream on another pair. The MAC provides nibble-wide (4-bit) data which is encoded into 5-bit code groups and inserted into the transmit data stream. The transmit packet is encapsulated by replacing the first two preamble nibbles with a start of stream delimiter (SSD) (/J/K codes) and appending an end of stream delimiter (/T/R codes) to the end of the packet. When the MAC indicates a transmit error, the transmit error code group (/H) is sent in place of the corresponding data code-group. The transmitter repeatedly sends the idle code-group between packets. The encoded data stream is serialized and then scrambled by the stream cipher block, as described later in this document. The scrambled data is then encoded into MLT3 signal levels.

In 1000BASE-T mode, the BCM57XX MAC Transceivers simultaneously transmit and receive a continuous data stream on all four twisted pairs in the CAT 5 cable. When a packet is pending transmission from the MAC, byte-wide data from the MAC is scrambled, trellis encoded into a four-dimensional code-group (a PAM5 symbol on each of the four twisted pairs), and inserted into the transmit data stream. The transmit packet is encapsulated by replacing the first two bytes of preamble with a start-of-stream delimiter and appending an end-of-stream delimiter to the end of the packet. When the MAC indicates a transmit error during a packet, a transmit error code-group is sent in place of the corresponding data code-group. The transmitter sends idle code-groups or carrier extend code-groups between packets. Carrier extension is used by the MAC to separate packets within a multiple packet burst. Carrier extend error is indicated by replacing the transmit data input with 1Fh during carrier extension.

Decoder

In 10BASE-T mode, Manchester decoding is performed on the data stream. In 100BASE-TX mode, the receive data stream, following equalization and clock recovery, is converted from MLT3 to serial NRZ data. The NRZ data is descrambled by the stream cipher block, as described later in this document. The descrambled data is then deserialized and aligned into 5-bit code-groups. The 5-bit code-groups are decoded into 4-bit data nibbles. The start-of-stream delimiter is replaced with preamble nibbles, and the end of stream delimiter and idle codes are replaced with 0h. The decoded data is provided to the MAC. When an invalid code-group is detected in the data stream, the BCM57XX MAC Transceivers indicate a receive error to the MAC. The receive error signal also asserts when the link fails, or when the descrambler loses lock during packet reception.

In 1000BASE-T mode, the receive data stream passes through a Viterbi decoder and descrambler and is translated back into byte wide data. The start of stream delimiter is replaced with preamble bytes and the end of stream delimiter and idle codes are replaced with 00h. Carrier extend codes are replaced with 0Fh or 1Fh. The decoded data is provided to the MAC. When an invalid code-group is detected in the data stream, the BCM57XX MAC Transceivers send an error condition to the MAC. The device also indicates receive error when receiving carrier extend code-groups, or when the local receiver status becomes unreliable during packet reception.

Carrier Sense

In 10BASE-T mode, carrier sense is recognized when valid preamble activity is detected on the TRD± input pins.

For 100/1000BASE-T operation, carrier sense is recognized as soon as activity is detected in the receive data stream. Data valid is recognized as soon as a valid SSD is detected. The end of the valid data is detected by the end of stream delimiter or two consecutive idle code-groups in the receive data stream. If activity is detected and a valid SSD is not detected immediately, the packet will be received with an error indication.

Link Monitor

In 10BASE-T mode, a link-pulse detection circuit constantly monitors the TRD± pins for the presence of valid link pulses.

In 100BASE-TX mode, receive signal energy is detected by monitoring the receive pair for transitions in the signal level. Signal levels are qualified using squelch detect circuits. When no signal is detected on the receive pair, the link monitor enters the Link Fail state, and transmission and reception of data packets is disabled. When a valid signal is detected on the receive pair for a minimum of one millisecond, the link monitor enters the Link Pass state, and the transmit and receive functions are enabled.

In 1000BASE-T mode, following auto-negotiation, the master transceiver begins sending data on the twisted-pair link. When the slave transceiver has recovered the master's timing, it also begins transmitting. Each end of the link continuously monitors its local receiver status. When the local receiver status has been OK for at least 1 µs, the link monitor enters the Link Pass state, and transmission and reception of data packets is enabled. When the local receiver status is not OK, and remains so for 750 ms, the link monitor enters the Link Fail state, and transmission and reception of data packets are disabled.

Digital Adaptive Equalizer

The digital adaptive equalizer removes intersymbol interference created by the transmission channel media. The equalizer accepts sampled unequalized data from the ADC on each channel and produces equalized data. The BCM57XX MAC-Transceivers achieve an optimum signal-to-noise ratio by using a combination of Feed Forward Equalization (FFE) and Decision Feedback Equalization (DFE). This powerful technique achieves a BER of less than 1×10^{-12} for transmission up to 100 meters on CAT 5 twisted-pair cable (100 meters on CAT 3 for 10BASE-T), even in harsh noise environments. The all-digital nature of the design makes the performance very tolerant to chip noise. The filter coefficients are self-adapting to any quality of cable or cable length.

Echo Canceller

Because of the bidirectional nature of the channel in 1000BASE-T, an echo impairment is caused by each transmitter. The echo canceller removes the transmitted signal impairment from the incoming receive signals. The output of the echo filter is added to the FFE output to cancel the echo impairment. The echo canceller coefficients are self-adapting to manage the varying echo impulse responses caused by different channels, transmitters, and environmental conditions.

Crosstalk Cancellers

Because the BCM57XX MAC Transceivers transmit and receive a continuous data stream on all four twisted-pair strands, the symbols sent by the other three local transmitters cause impairments on the received signal for each channel through near-end-crosstalk (NEXT) mechanism between the pairs. Because each receiver has access to the data for the other three pairs that cause this interference, it is possible to cancel the effect. This is accomplished with three adaptive NEXT cancelling filters. The outputs of these filters are added to the FFE output to cancel the NEXT impairment.

Analog-to-Digital Converter

Each receive channel has its own 125-MHz Analog-to-Digital Converter (ADC). The ADC samples the incoming data on the receive channel and feeds the output to the digital adaptive equalizer. Advanced analog circuit techniques achieve low offset, high-power supply noise rejection, fast settling time, and low bit-error rate.

Clock Recovery/Generator

The clock recovery and generator block creates the 125-MHz transmit and receive clocks for 1000BASE-T, 100BASE-TX, and 10BASE-T operation. In 10BASE-T or 100BASE-TX mode, the transmit clock is locked to the 25-MHz crystal input, and the receive clock is locked to the incoming data stream. In 1000BASE-T mode, the two ends of the link perform loop timing. One end of the link is configured as the master and the other end as the slave. The master transmit and receive clocks are locked to the 25-MHz crystal input. The slave transmit and receive clocks are locked to the incoming receive data stream. Loop timing allows for cancellation of echo and NEXT impairments by insuring that the transmitter and receiver at each end of the link are operating at the same frequency. The phase relationship between the transmit and receive clocks is determined by the round-trip delay of the loop. The BCM57XX MAC Transceiver timing recovery circuits track any changes in the relative transmit and receive phases caused by timing jitter.

Baseline Wander Correction

The 1000BASE-T and the 100BASE-TX data streams are not always DC-balanced. Because the receive signal must pass through a transformer, the DC offset of the differential receive input can wander. This effect, known as baseline wander, can greatly reduce the noise immunity of the receiver. The BCM57XX MAC Transceivers automatically compensate for baseline wander by removing the DC offset from the input signal and thereby significantly reducing the probability of a receive symbol error.

In 10BASE-T mode, baseline wander correction is not performed because the Manchester coding provides perfect DC balance.

Multimode TX DAC

The multimode transmit Digital-to-Analog Converter (DAC) transmits PAM5, MLT3, and Manchester coded symbols. The transmit DAC performs signal wave shaping, which decreases the unwanted high-frequency signal components, thus reducing EMI. The transmit DAC utilizes a current drive output which is well-balanced and produces very low noise transmit signals.

STREAM CIPHER

In 1000BASE-T and 100BASE-TX modes, the transmit data stream is scrambled to reduce radiated emissions and to ensure that there is no correlation between symbols in the data stream. The 1000BASE-T scrambler also ensures that there is no correlation between symbols on the four different wire pairs, and that there is no correlation between symbols in the transmit and receive data streams. The scrambler reduces peak emissions by randomly spreading the signal energy over the transmit frequency range and eliminating peaks at certain frequencies. The randomization of the data stream also assists the digital adaptive equalizers and echo/crosstalk cancellers. The adaptive algorithms in these circuits require that there to be no sequential or cross-channel correlation between symbols in the various data streams.

In 100BASE-TX mode, the transmit data stream is scrambled by exclusive ORing the encoded serial data stream with the output of an 11-bit wide Linear Feedback Shift register (LFSR), which produces a 2047-bit non-repeating sequence. In 1000BASE-T mode, the GMII transmit data is scrambled by exclusive-ORing the input data byte with an 8-bit wide cipher text word. The cipher text word is generated each symbol period from eight uncorrelated maximal length data sequences, which are produced by linear remappings of the output of a 33-bit wide LFSR. The scrambled data bytes are encoded, and then the sign of each transmitted symbol is again randomized by a 4-bit wide cipher text word, which is generated in the same manner as the 8-bit word. The master and slave transmitters use different scrambler sequences to generate the cipher text words.

The receiver descrambles the incoming data stream by exclusive-ORing it with the same sequence generated at the transmitter. The descrambler detects the state of the transmit LFSR by seeking a sequence representing consecutive idle code-groups. The descrambler locks to the scrambler state after detecting a sufficient number of consecutive idle codes. The BCM57XX MAC Transceivers only enable transmission and reception of packet data when the descrambler is locked. The receiver continually monitors the input data stream to ensure that it has not lost synchronization by checking that inter-packet gaps containing idles or frame extensions are received at expected intervals. When the BCM57XX MAC-Transceivers detect loss of synchronization, it notifies the remote PHY of the inability to receive packets (1000BASE-T mode only) and attempt to resynchronize to the received data stream. If the descrambler is unable to resynchronize for a period of 750 ms, the BCM57XX MAC Transceivers are forced into the Link Fail state.

In 10BASE-T mode, scrambling is not required to reduce radiated emissions.

WIRE MAP AND PAIR SKEW CORRECTION

During 1000BASE-T operation, the BCM57XX MAC Transceivers have the ability to automatically detect and correct some UTP cable wiring errors. Wiring errors caused by swapping of pairs within the UTP cable, as well as polarity errors caused by swapping of the wires within a pair, are detected by the symbol decoder and compensated for internally to the BCM57XX MAC Transceivers. The BCM57XX MAC Transceivers also automatically compensate for differences in the arrival times of symbols on the four pairs of the UTP cable. The varying arrival times are caused by differing propagation delays (commonly referred to as delay skew) between the wire pairs. The BCM57XX MAC Transceivers can tolerate delay skews up to 120 ns.

During 10/100 operation, pair swaps are corrected. Delay skew is not an issue because only one pair is used in each direction.

Auto-Negotiation

The BCM57XX MAC Transceivers negotiate its mode of operation over the twisted-pair link using the auto-negotiation mechanism defined in the IEEE 802.3u and 802.3ab specifications. Auto-negotiation can be enabled or disabled by hardware or software control. When the auto-negotiation function is enabled, the BCM57XX MAC Transceivers automatically choose the mode of operation by advertising its abilities and comparing them with those received from its link partner. The BCM57XX MAC Transceivers can be configured to advertise 1000BASE-T full-duplex and/or half-duplex, 100BASE-TX full-duplex and/or half-duplex, and 10BASE-T full-duplex and/or half-duplex. The transceiver negotiates with its link partner and chooses the highest operating speed. Auto-negotiation can be disabled for testing or for 100BASE-TX or 10BASE-T operation, but is always required for normal 1000BASE-T operation. For details on auto-negotiation using Next Page exchange.

Automatic MDI Crossover

During auto-negotiation, one end of the link must perform an MDI crossover so that each transceiver's transmitter is connected to the other receiver. The BCM57XX MAC Transceivers can perform an automatic MDI crossover when the Disable Automatic MDI Crossover bit in the PHY Extended Control register (see "PHY Extended Control Register (PHY_Addr = 0x1, Reg_Addr = 10h)" on page 623) is disabled, thus eliminating the need for crossover cables or cross-wired (MDIX) ports. During auto-negotiation, the BCM57XX MAC Transceivers normally transmit on TRD±{0} and receive on TRD±{1}. When connected to another device that does not perform the MDI crossover, the BCM5701 MAC Transceiver automatically switches its transmitter to TRD±{1} and its receiver to TRD±{0} to communicate with the remote device. If two devices that both have MDI crossover capability are connected, an algorithm determines which end performs the crossover function. During 1000BASE-T operation, the BCM57XX MAC Transceivers swap the transmit symbols on pairs 0 and 1 and pairs 2 and 3 if auto-negotiation completes in the MDI crossover state. The 1000BASE-T receiver automatically detects pair swaps on the receive inputs and aligns the symbols properly within the decoder.

Wire Speed

Wire Speed is a mode which controls the auto-negotiation advertising. At startup, all the 100BASE-T (TX) and 1000BASE-T (Gigabit) modes selected in MII registers 4 and 9 are advertised. If the same HCD fails to link five consecutive times, then the highest ability not already masked out will be masked out. When both TX and Gigabit abilities are masked out and the same HCD fails five consecutive times, then all the advertised abilities in MII register 4 and 9 are advertised. Whenever TX and Gigabit is masked out, then the Wire Speed down grade is active, indicating that not all the requested advertised abilities are actually advertised to the Link Partner.

Note: The 10BASE-T is always advertised as requested per MII register 4 at all times and is not affected by the Wire Speed mode. Also, the Wire Speed mode will never mask out Gigabit or TX abilities unless there are other abilities available to advertise.

When Wire Speed mode is enabled, the HCD can be determined from MII register 19h bits 10:8 since not all of MII register 4 advertised abilities may be sent to the Link Partner when down grade is active as shown by a 1 in bit 14 of MII register 11h.

PHY CONTROL

The BCM57XX family supports three physical layer interfaces:

- The MII is used in conjunction with 10/100 Mbps copper Ethernet transceivers.
- GMII supports 1000 Mbps copper Ethernet transceivers.
- The TBI connects to a Serializer/Deserializer (SerDes) for 1000 Mbps optical transceivers.

MII BLOCK

The MII interconnects the MAC and PHY sublayers (see the following figure).

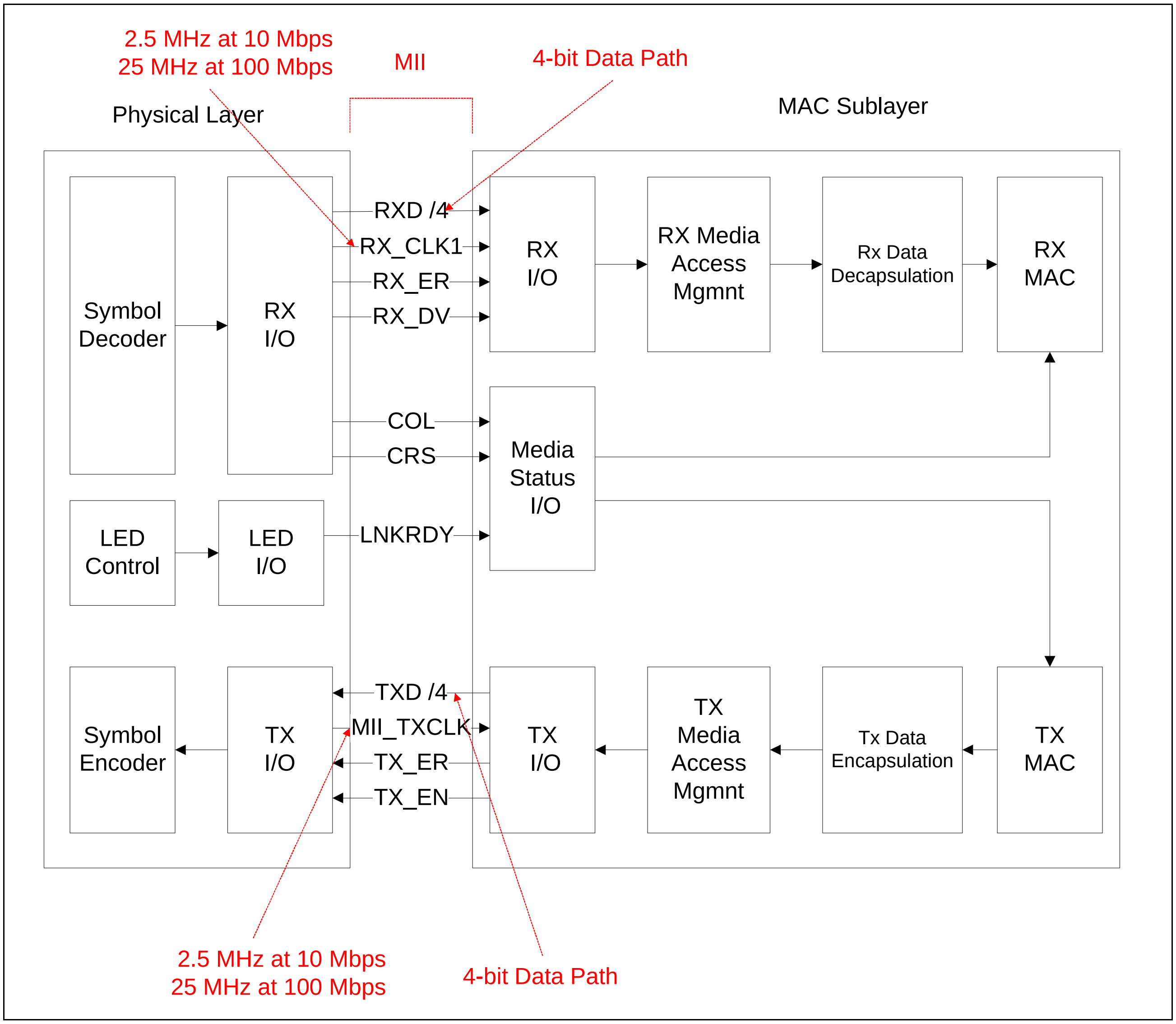


Figure 32: Media Independent Interface

The specifics of MII may be located in section 22 of the IEEE 802.3 specification. RXD[0..3] are the receive data signals; TXD[0..3] are the transmit data signals. MII operates at both 10 Mbps and 100 Mbps wire-speeds. (Gigabit Ethernet uses the GMII standard.) When MAC and PHY are configured for 10 Mbps operation, the RX_CLK1 and MII_TXCLK clocks run at 2.5 MHz. Both RX_CLK1 and MII_TXCLK are sourced by the PHY. 100 Mbps wire speed require RX_CLK1 and MII_TXCLK to provide a 25-MHz reference clock. Receive Data Valid (RX_DV) is asserted when valid frame data is received; at any point during data reception, the PHY may assert Receive Error (RX_ER) to indicate a receive error. The MAC will record this error in the statistics block. The MAC may discard a bad RX frame—exception being sniffer/ promiscuous modes (see Allow_Bad_Frames bit in MAC mode register). The Transmit Enable (TX_EN) signal is asserted when the MAC presents the PHY with a valid frame for transmission. The MAC may assert TX_ER to indicate the remaining portion of frame is bad. The PHY will insert Bad Code symbols into the remaining portion of the frame. A detected collision in half-duplex mode may be such a scenario where TX_ER is asserted. The PHY will assert COL when a collision is detected. The COL signal is routed to both the RX and TX MACs. The transmit MAC will back off transmission and the RX MAC will throw away partial frames.

The 10 Mbps physical layer uses Differential Manchester encoding on the wire. Manchester encoding uses two encoding levels: 0 and 1. 100 Mbps Ethernet requires MLT-3 waveshaping on the transmission media. MLT-3 uses three encoding levels: -1, 0, and 1. Both physical signaling protocols are transparent to the MAC sublayer and are digitized by the PHY. The PHY encodes/decodes analog waveforms at its lower edge while the PHY presents digital data at its upper edge (MII).

GMII Block

The GMII is full-duplex (see Figure 33 on page 71); the send and receive data paths operate independently.

The transmit signals TXD[0..7] create a eight-bit wide data path. The TXD[0..7] signals are synchronized to the reference clockTX_CLK0. The TX_CLK0 clock runs at 125 MHz and is sourced by the MAC sublayer. Transmit Error (TX_ER) is asserted by the MAC sublayer. The PHY will transmit a bad code until TX_ER is de-asserted by the MAC. TX_ER is driven synchronously with TX_CLK0. The Transmit Enable (TX_EN) indicates that valid data is presented on the TXD lines. The TXD[0..7] data is framed on the rising edge of TX_EN.

The receive data path is also eight bits wide. RXD[0.7] are sourced by the PHY. When valid data is presented to the MAC sublayer, the PHY will also assert Receive Data Valid (RX_DV). The rising edge of RX_DV indicates the beginning of a frame sequence. The PHY drives the reference clock RX_CLK1, so the MAC sublayer can synchronize data sampling on RXD[0..7]. The PHY may assert RX_ER to indicate frame data is invalid; the MAC sublayer must consider frames in progress incomplete.

When the MAC operates in half-duplex mode, a switch or node may transmit a jamming pattern. The PHY will drive the Collision (COL) signal so the MAC may back off transmission and throw away partially received packet(s). The COL signal will also cause the TX MAC to stop the transmission of a packet. The COL signal is not driven for full-duplex operation since collisions are undefined. The PHY will drive Carrier Sense (CRS) as a response to traffic being sent/received. The MAC sublayer can monitor traffic and subsequently drive traffic LEDs. Finally, LNKRDY may be driven from the PHY's link LEDs. The routing of signals from the PHY is application specific. The BCM57XX family reference NIC routes the PHY 10 Mbps link signal to the LNKRDY pin on the BCM57XX family. The MAC may poll LNKRDY in 10 Mbps mode to determine if link is present.

Pulse Amplitude Modulated Symbol (PAM5) encoding is leveraged for Gigabit Ethernet wire transmissions. PAM5 uses five encoding levels: -2, -1, 0, 1, and 2. Four symbols are transmitted in parallel on the four twisted-wire pairs. The four symbols create a code group (an eight-bit octet). The process of creating the code-group is called 4D-PAM5. Essentially, eight data bits are represented by four symbols. Table 40-1 in the 802.3ab specification shows the data bit to symbol mapping. The code group representation is also referred to as a quartet of quinary symbols {TA, TB, TC, TD}. The modulation rate on the wire is measured at 125 Mbaud. The resultant bandwidth is calculated by multiplying 125 MHz by eight bits, for 1000 Mbps wire speed.

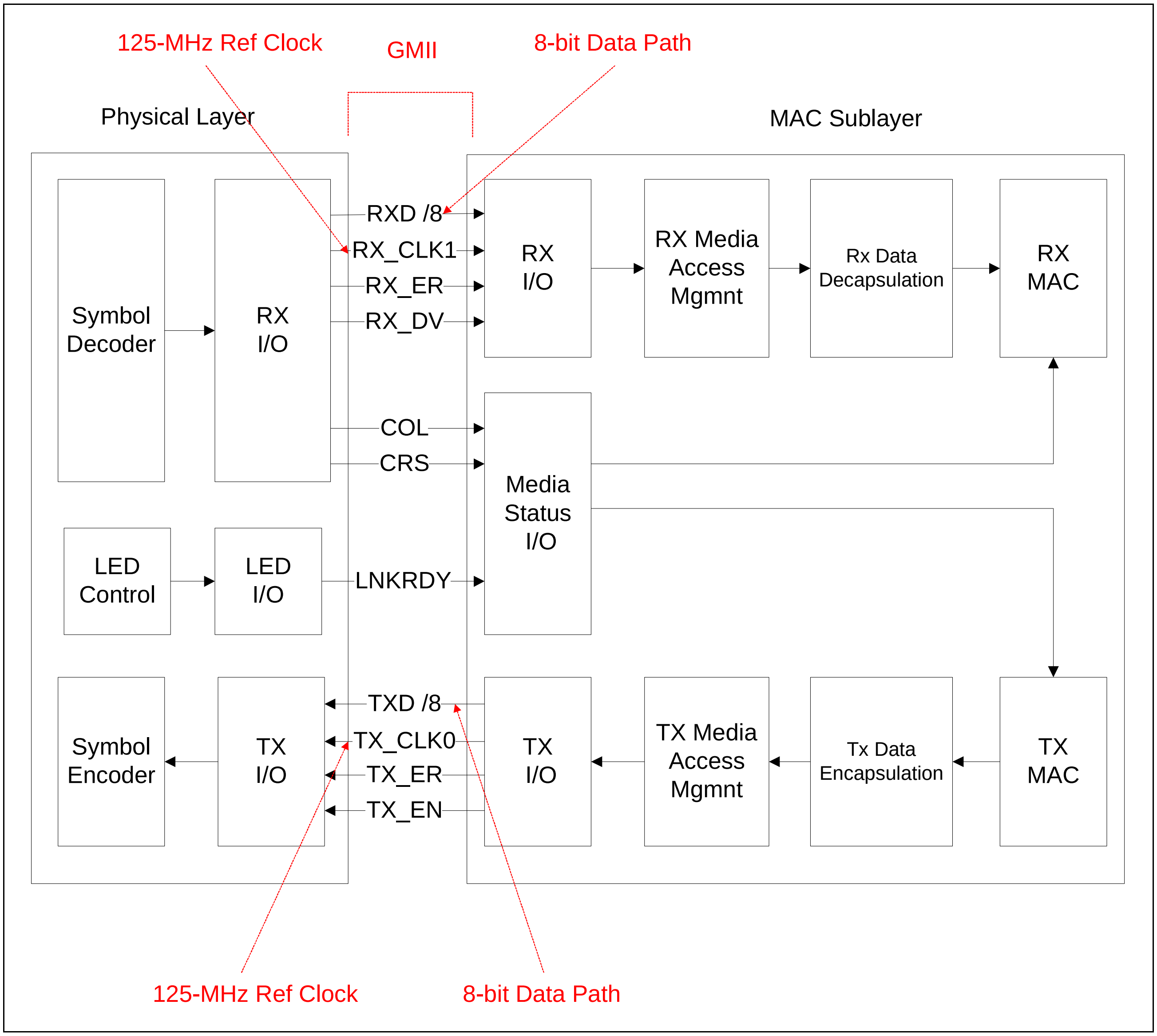


Figure 33: GMII Block

TBI Block

The Serializer Deserializer (SerDes) device bridges a parallel TBI to a serial Pseudo Emitter Coupled Logic (PECL) interface. Gigabit optical transceivers convert multimode/single-mode laser energy to an electrical waveform—a 1.25-GHz PECL signal. The specifics of the Gigabit fiber communications can be located in the IEEE 802.3z specification. The PECL interface is full-duplex and the clocking is interleaved with data.

In Figure 34, TDn represents the PECL transmit interface and RDn represents the PECL receive interface. Again, TDn and RDn are 1.25-GHz serial PECL streams. The SerDes device converts both TDn and RDn signals to the parallel TBI signals TXD[0..9] and RXD[0..9]. TBI is commonly called 8B/10B encoding. TBI does not embed clocking in the data streams; rather transmit and receive data streams are synchronized with clocking signals—TX_CLK0, RX_CLK0, and RX_CLK1. The transmit clock TX_CLK0 runs at 125 MHz and is asymmetric to the receive clocking. The receive clocking mechanism is slightly different from the transmit side—two 62.5-MHz clocks are required. RX_CLK0 runs at 62.5 MHz and is 180 degrees out of phase from the 62.5 RX_CLK1 signal. The receive/transmit data paths are 10 bits wide, with a 125 MHz data transfer rate. The resultant bandwidth of each connection is 1.25 GHz. The RXD and TXD connections are full-duplex.

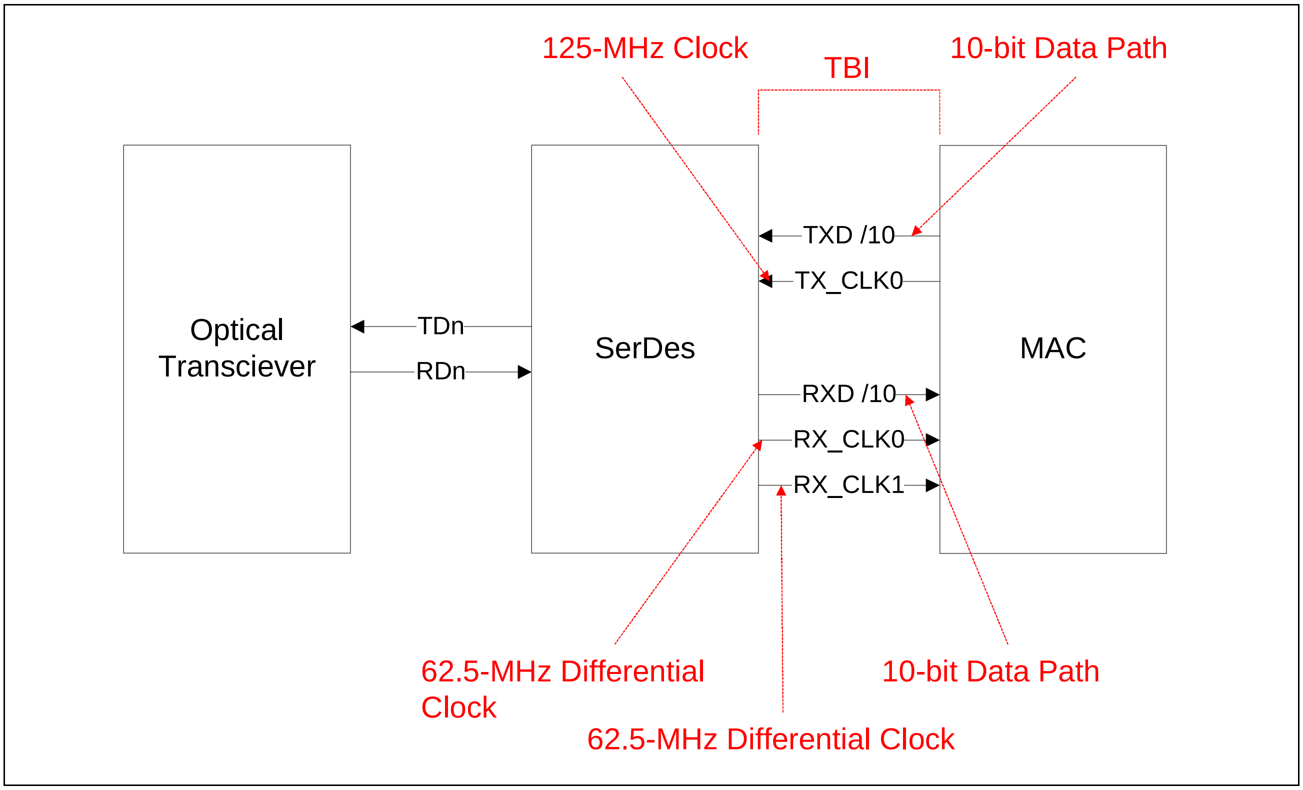


Figure 34: TBI Block

MDIO Register Interface

The following figure shows the MDI register interface.

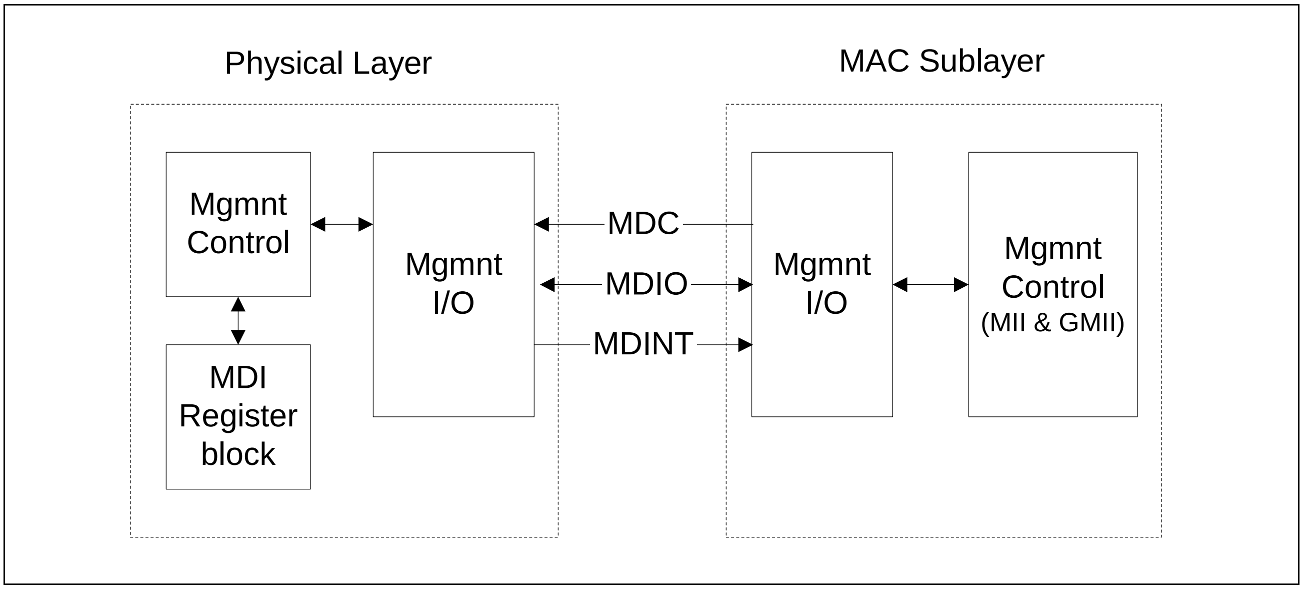


Figure 35: MDI Register Interface

Management Data Clock

The Management Data Clock (MDC) is driven by the MAC sublayer. The PHY will sink this signal to synchronize data transfer on the MDIO signal—MDC is a reference clock. This clock is not functionally associated to either RX_CLK or TX_CLK. The minimum period for this clock is 400 ns with high and low times having 160 ns duration.

Management Data Input/Output

The Management Data Input/Output (MDIO) signal passes control and status data, between the MAC and PHY sublayers. MDIO is a bidirectional signal, meaning both the PHY and MAC may transfer data. The MAC typically transfers control information and polls status; whereas, the PHY transfers status back to the MAC, using MDIO.

Management Data Interrupt

The Broadcom PHY (i.e., BCM5401) may be programmed to generate interrupts. A PHY status change initiates a Management Data Interrupt (MDINT). A MDI mask register allows host software to selectively enable/disable status types, which cause MDINT notification. The PHY will assert $\overline{\text{INTR}}$ until software clears the interrupt. Reading the status register will clear the interrupt.

Management Register Block

The layout and configuration of MDI register block is device dependent. The MDI register block is the control/status access point, which host software may read/write. The IEEE 802.3 specification defines a basic register block for MII and GMII; the basic register set contains control and status registers. GMII also exposes an extended register set, used in 1000 Mbps configuration/status. Refer to the specific PHY for details on the register layout (refer to the BCM5401 data sheet). The fundamentally point is to understand that the MDC and MDIO signals are used to access the MDI register block.

UNIVERSAL MANAGEMENT PORT (APPLICABLE TO BCM5714C/ BCM5714S/BCM5715C/BCM5715S ONLY)

The BCM5714C/BCM5714S/BCM5715C/BCM5715S devices support a UMP interface that allows devices such as the Baseboard Management Controller (BMC) to send and receive system management information through the Gigabit Ethernet ports. This eliminates the need for a dedicated Ethernet port and reduces system cost. The UMP interface is a simple MII/RMII Ethernet connection which always runs in full-duplex 100 Mbit mode and uses pause frames to support flow control between the UMP nodule and the baseboard management controller (BMC).

The BCM5714 supports two UMPs which share the same physical interface to an external device such as a BMC. The UMP interface on the BCM5714 functions similarly to a MII or a Reduced MII (RMII), but does not support the same number of pins as in either interface. Unless stated otherwise, the MII/RMII mentioned throughout this section refer to the UMP interface unique to the BCM5714, not the standard MII/RMII commonly seen in a MAC-to-PHY interface.

The UMP interface is designed to support up to four UMPs. The UMP address is selected by strapping the PWR_IND pin. The BMC addresses individual UMPs using the two ID bits embedded in the control command. The UMP passes data packets from the BMC to the Ethernet MAC for transmission. Upon receiving packets addressed to the BMC, the Ethernet MAC routes the receive packets to the UMP, which in turn forwards these packets to the BMC. Each UMP delivers 20 Mbps or more sustained throughput when the physical layer is connected at 1 Gbps and 8 Mbps or more (typically 20 Mbps) when the physical layer is connected at 100 Mbps.

Since the signals from multiple UMP devices are hardwired together, the output of the transmit signals are normally open-drained with a weak internal pull down resistor. The transmit signals are only driven when a UMP device sends data.

The MII contains all the signals required to transmit and receive at 100 Mbps in full-duplex mode. The transmit signals include TXD[3:0], TX_ER, TX_EN, and TX_CLK. The receive signals include RXD[3:0], RX_DV, RX_ER, and RX_CLK. Both TX_CLK and RX_CLK are clock input signals which operate at a maximum frequency of 25 MHz. The BCM5714C provides a 25-MHz clock signal (CK25), that can be used to drive TX_CLK and RX_CLK. Any other clock frequency will require an external clock source.

The RMII provides the same bandwidth as the MII, but uses half the number of pins. The RMII includes seven signals: TXD[1:0], TX_EN, RXD[1:0], RX_DV, and REFCLK. The reference clock (REFCLK) operates at any frequency between 2.5 MHz and 50 MHz. The BCM5714 provides a 25-MHz clock signal (CLK25) that can be used to drive REFCLK. Any other clock frequencies will require an external clock source.

Figure 36 and Figure 37 show how two BCM5714Cs are connected to one BMC via the MII and RMII.

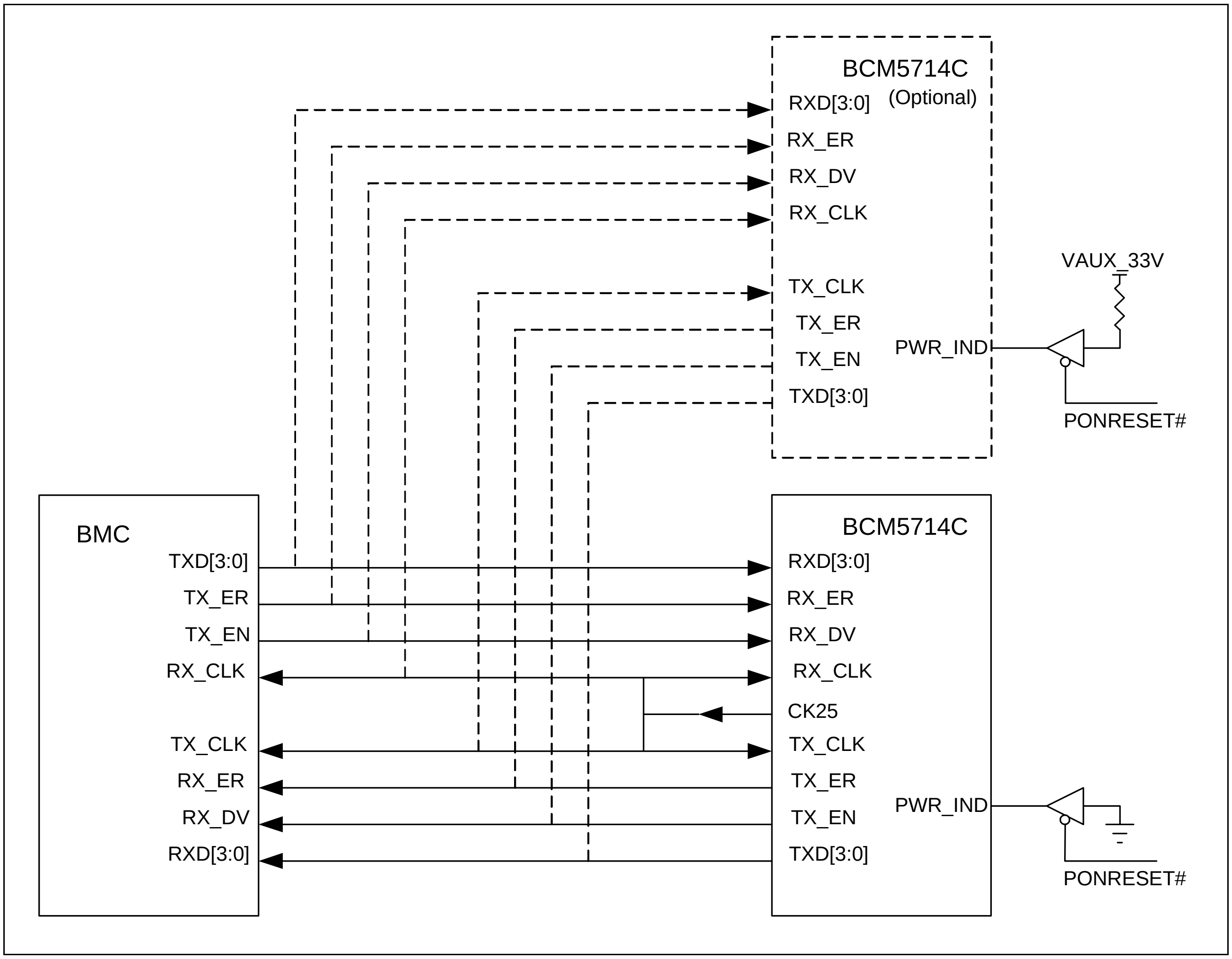


Figure 36: UMP MII Connections Between BMC and UMP Supported BCM57XX Devices

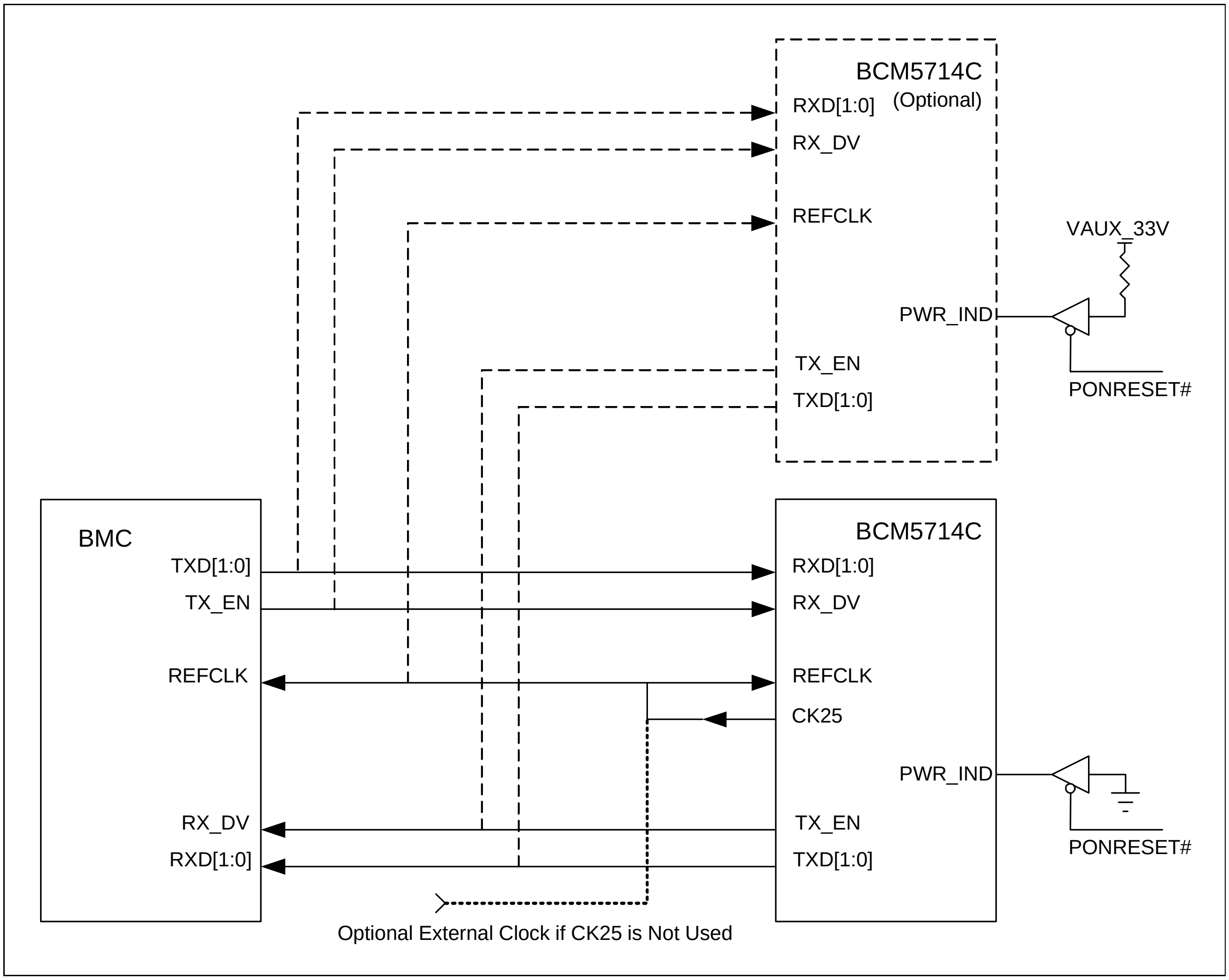


Figure 37: UMP RMII Connections Between BMC and UMP Supported BCM57XX Devices

UMP Receive Data Path

1. All incoming frames on the UMP interface are examined by the UMP MAC. Invalid frames are discarded and valid frames are stored in the UMP RX FIFO.
2. The UMP receive attention is generated to notify the CPU of the received packet.
3. The CPU examines the received packet and takes action if it is the addressee. The action taken depends on the command enclosed in the packet. If it must transfer the packet to the Gigabit MAC, the succeeding steps occur.
4. The CPU transfers the data into TX MBUFs of the gigabit transmit path and writes an entry into gigabit TX MAC.
5. The gigabit TX MAC processes this entry and sends out the packet on GMII.

UMP Transmit Data Path

1. The incoming frames on the gigabit interface are stored into RX MBUFs.
2. The MAC destination address of the received packet is compared with the addresses that are programmed in the UMP filter module of the gigabit MAC. If a match occurs, then the RDI module generates attention to the CPU.
3. The CPU queries the UMP TX module to see if it can accept a new packet. If space is available, then it transfers the packet in to the UMP transmit FIFO.
4. The UMP MAC then transmits the frame to the IMD through the UMP transmit interface.

UMP Programming Details

The UMP Attention Enable register 0x7800 allows us to enable various attentions (Enable TX Ready Event, Enable RX Ready Event,…) from UMP module. When any of the enabled events happen, the UMP module generates the attention through the UMP Attention Status register 0x7804. The UMP firmware which responds to the UMP software commands uses the UMP registers 0x7810-0x782C. The UMP firmware configures the UMP registers depending upon the configuration commands received from the BMC and maintains the configuration across device resets. Refer to the UMP specification application note for various UMP commands supported by UMP firmware.

UMP Filter Module

Any packet that is received on the gigabit interface and which is not filtered by the L2 address filter logic is examined by the UMP filter block. The function of the UMP filter block is to check if the incoming packet matches the UMP filter rules programmed by the CPU. If a match occurs, then the UMP filter block resets the pass bit in the RDI FTQ which results in RDI-FTQ attention being generated to the CPU.

The two fields in the incoming frame that are checked by the UMP filter module are the destination address and VLAN-ID field. The algorithm that is used to filter the incoming packets and generate the attention to the CPU is described below.

```
Ump_rdi_attn = (((incoming_vlan_tag == ump_vlan_tag) && ump_vlan_cmp_en)
                             || ! Ump_vlan_cmp_en)  &&
                             (incoming_mac_da == broadcast_addr && ump_bc_cmp_en) ||
                             (incoming_mac_da == ump_mc_addr  && ump_mc_cmp_en) ||
                             (incoming_mac_da == gig_mac0_addr && mac0_cmp_en) ||
                             (incoming_mac_da == gig_mac1_addr && mac1_cmp_en) ||
                             (incoming_mac_da == gig_mac2_addr && mac2_cmp_en) ||
                             (incoming_mac_da == gig_mac4_addr && mac3_cmp_en);
```

The ump_vlan_tag (register 0x5E8), the UMP multicast address (ump_mc_addr, register 0x5D8, MC Address Mask register is 0x5E4) and the compare enable bits (register 0x5F0) are programmable.

The UMP filter module has the lowest priority between the L2 address filter and the rule checker modules. If the ruler checker and the UMP filter modules are programmed to have contradictory rules, then the UMP filter output is overridden by the rule checker. For example, if RC is programmed to drop a broadcast frame and UMP filter module is programmed to generate an attention for a broadcast frame then the broadcast frame is dropped. Similarly, the L2 address filter has higher priority than the UMP filter logic. For example, if the L2 address filter's multicast hash table is programmed and if the UMP filter module's Multicast frame address does not fall in the MC hash table, then the MC frame destined for the UMP will not make it through. In order to overcome this limitation, the UMP filter logic has control bit (bit 7 of register 0x5F0) to disable the MC hash function of the L2 address filter.

RDI TIMER ATTENTION

The BCM5714 and BCM5715 devices support RDI Timer Attention to detect the condition of PCI bus hung (driver blue screen scenario). The RDI timer module generates an attention if entries from the RDI FTQ are not being de-queued for a certain period of time. Enable the RDI Timer Attention by setting the bit 7 of RDI Mode register (0x2400). Program the Time Out value for generating RDI Timer Event Attention in bits 5:4 of RDI Timer Mode register (x24F0). The FW can check the bit 7 of RDI Status register (0x2404) for RDI Timer Attention to recognize the blue screen scenario and to take the appropriate action.

PCI/PCI-X Interface

The BCM57XX family has a PCI v2.2 and PCI-X v1.0 compliant bus interface (see the following figure).

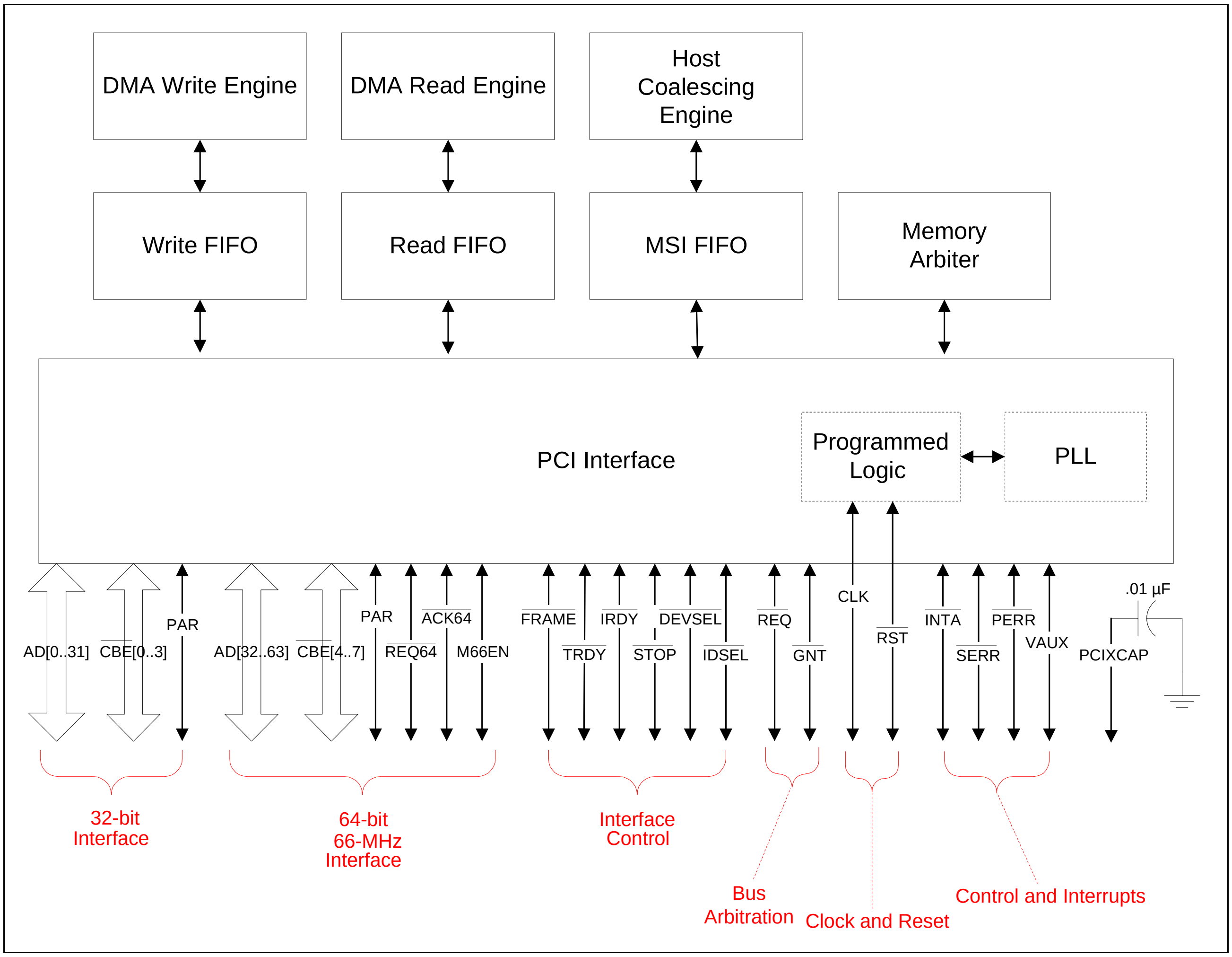


Figure 38: PCI/PCI-X Bus Interface

The remaining pins can be broken into the following categories:

- 32-bit interface—pins necessary for 32-bit PCI compatibility
- 64-bit extension—pins necessary for 64-bit bus compatibility
- Clock and reset—pins necessary for transaction timing and device reset
- Interface control—pins necessary for PCI bus master, configuration, and bus target transactions
- Bus arbitration—pins necessary to request and grant PCI bus
- Control and interrupt—pins necessary to generate/sample interrupts, bus errors, and voltage levels

Dual Address Cycle

Both the PCI and PCI-X bus interfaces support Dual Address Cycle (DAC). The BCM57XX family supports DAC as both a master and a target. The device may bus-master data from physical addresses greater than 4 gigabytes using DAC mode. The device will not generate a DAC transaction unless a host programmed control block specifies a physical address greater than 4 gigabytes. For example, a send ring buffer descriptor located above 0xFFFFFFFF requires a DAC read transaction (see "Common Data Structures" on page 89 for discussion of rings). DAC is not used for an address of 0xFFFFFFFF. DAC is used for addresses of 0x100000000 or higher. The address phase of a DAC PCI/PCI-X transaction requires two address cycles to be driven on AD[0..31]. The BCM57XX family does not know if the target device supports 64-bit addressing; thus, the AD[0..31] are used to source a high and low address. If the target device does support 64-bit addressibility, the BCM57XX family still drives the high address on AD[32..64]. There is no penalty for sourcing a high and low address on AD[0..31] even if the target supports 64-bit addressing; the target still requires one PCI_CLK to decode the address driven on the bus. The decode clock cycle is used to drive the high address on AD[0..31] and on the next clock DEVSEL is asserted. If the target is a 32-bit device, DEVSEL will not be asserted until at least one clock after the high address is driven on AD[0..31]. The following figure shows a read transaction with the DEVSEL being asserted based on target type (32 vs. 64 bit).

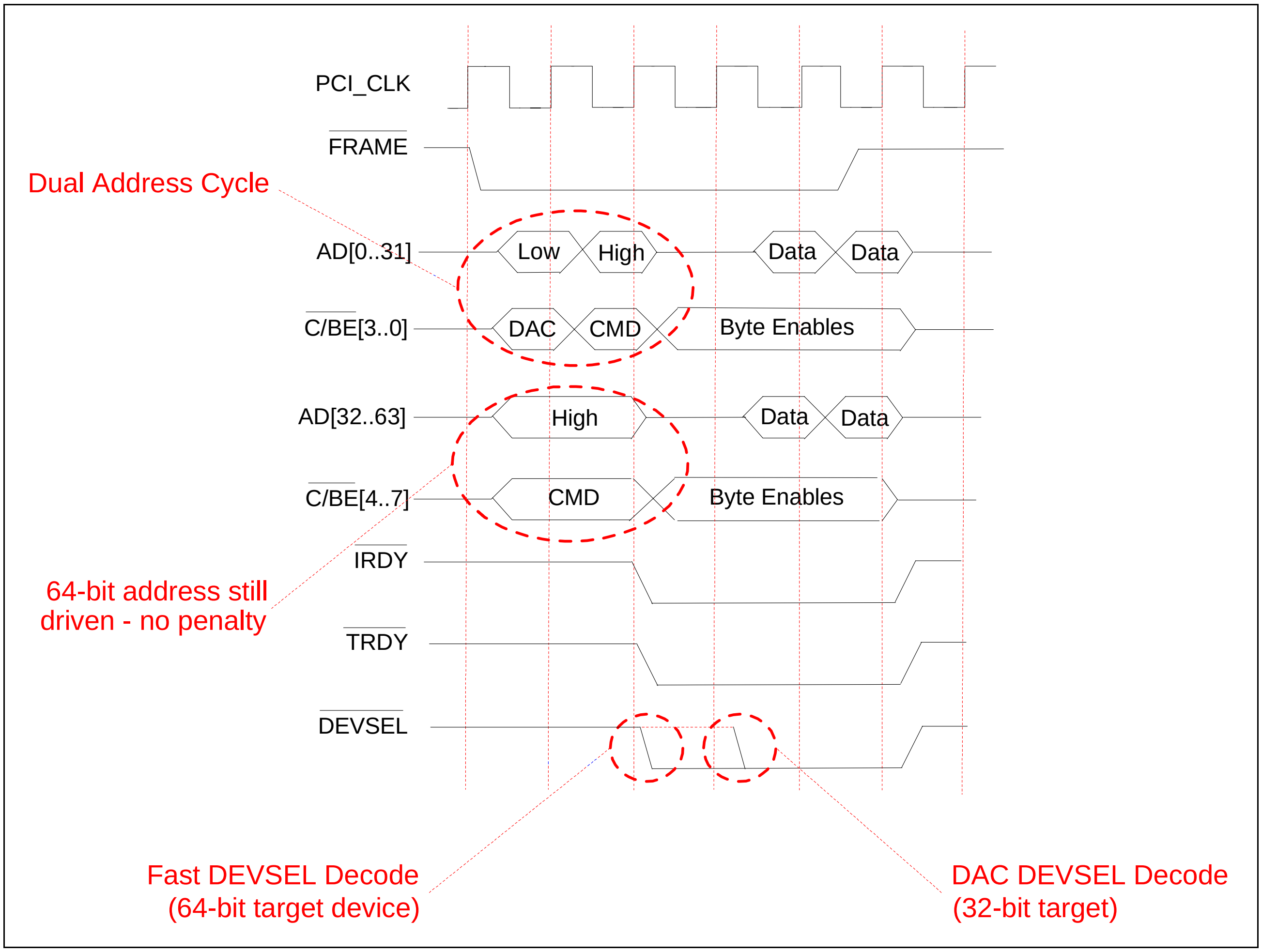


Figure 39: Read Transaction Based on Target Type

TARGET READ/WRITE BURSTS

When the BCM57XX family is a PCI Target, the PCI and PCI-X interfaces do not support burst read/write transactions. The device will claim the burst transaction, but will disconnect after the first data phase. The BCM57XX family will issue a single data phase disconnect; the transaction is subsequently terminated. $\overline{\text{DEVSEL}}$ deasserted on the data phase. Then $\overline{\text{TRDY}}$ and $\overline{\text{STOP}}$ will be asserted to indicate a Single Data Phase Disconnect. Finally, the initiator (master) releases the bus by deasserting $\overline{\text{FRAME}}$ and $\overline{\text{IRDY}}$. Refer to section 2.11.2.1 in the PCI-X v1.0 specification. The PCI v2.2 specification refers to this behavior as *disconnect with data*. The following figure shows a single data phase disconnect.

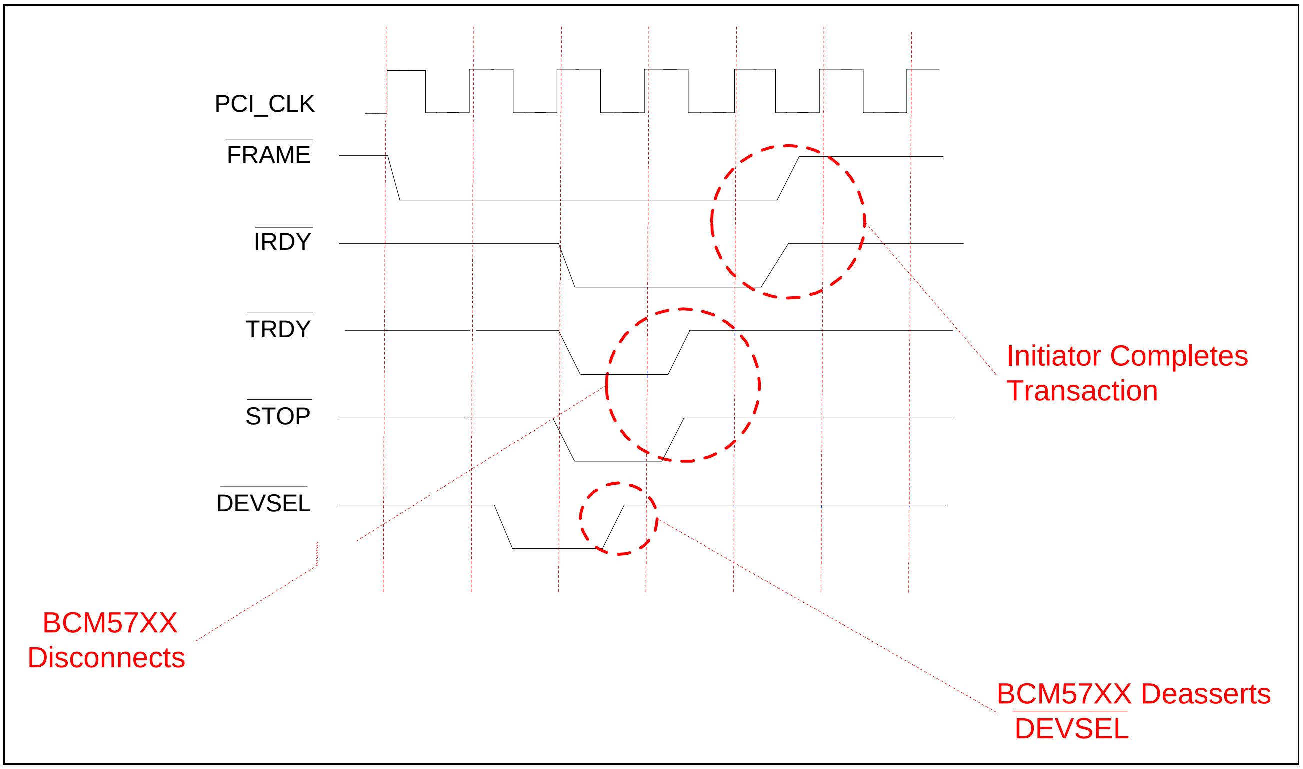

PCI_CLK
$\overline{\text{FRAME}}$
$\overline{\text{IRDY}}$
$\overline{\text{TRDY}}$
$\overline{\text{STOP}}$
$\overline{\text{DEVSEL}}$
Initiator Completes Transaction
BCM57XX Disconnects
BCM57XX Deasserts $\overline{\text{DEVSEL}}$

Figure 40: Single Data Phase Disconnect

PCI Host Bus Interface (Not Applicable to BCM5721, BCM5751, BCM5752, BCM5714C, BCM5714S, BCM5715C, and BCM5715S)

The BCM57XX family supports 32 and 64-bit PCI transactions, and the PCI interface runs at bus frequencies from 33 MHz to 66 MHz. The minimum tolerance for the PCI clock is 27 MHz. The BCM57XX family isolates the PCI clock from other chip clock domains and the MAC core does not drive off the PCI bus clock. A Phased Locked Loop (PLL) is embedded in the PCI interface to match the PCI bus frequency. This PLL allows the device to meet the timing requirements for PCI read/write transactions and minimize bus latency (wait states). When the device comes out of reset, the PLL samples and matches the PCI bus clock frequency. The PLL's configuration is also influenced by host software. The host programmable block affects the PLL speed based on the configuration of the 32_Bit_PCI_Bus and 33/66 MHz_PCI_66/133 MHz_PCIX bits in the PCI_State register (see "PCI State Register" on page 202). The host programmable block affects the PLL speed—BCM57XX firmware may reprogram the bus speed of the device using the 33/66 MHz_PCI_66/133 MHz_PCIX bit in the PCI_State register.

The PCI interface supports all transactions except I/O commands. The BCM57XX family does not have I/O mapped, thus the device will not respond to I/O read/write transactions. The following table lists the commands supported by the BCM57XX family.

Table 23: PCI Commands Supported by the* BCM57XX *Family

Command Encoding	Description	Usage
0110	Memory Read	BCM57XX family as PCI master or target[a]
0111	Memory Write	BCM57XX family as PCI master or target
1010	Configuration Read	BCM57XX family as PCI target
1011	Configuration Write	BCM57XX family as PCI target
1100	Memory Read Multiple	BCM57XX family as PCI master or target[a]
1101	Dual Address Cycle	BCM57XX family as PCI master
1110	Memory Read Line	BCM57XX family as PCI master or target

[a] When using the BCM57XX family as a PCI target, break after one data cycle (i.e., for register space, a maximum of 32 bits; for memory and mailbox space, a maximum of 64 bits).

PCI-X Host Bus Interface (Applicable to BCM5700, BCM5701, BCM5703C, BCM5703S, BCM5704C, and BCM5704S)

The BCM57XX PCI-X interface is pin for pin compatible with 64-bit PCI connectors; the PCI-X v1.0 specification is backward compatible with the PCI v2.2 standard. The host chipset uses a three level comparator to sample the NIC's PCIXCAP pin. Refer to table 6-1 in the PCI-X v1.0 specification. The Device PCI-X interface supports bus speeds ranging from 66 MHz to 133 MHz. For example, the BCM57XX PCI-X interface runs at 100-MHz bus frequency since some early PCI-X chipsets drive PCI_CLK to 100 MHz. The PCI-X interface will scale its clocking to 100 MHz, but the PCI-X status registers will indicate a 133 MHz bus frequency. Essentially, a PCI-X bus speed greater than 66 MHz will be programmable via 133 MHz status and control. The number of physical connectors (loads) a host chipset can drive is scaled to bus frequency (see the following table).

Table 24: PCI-X Bus Speeds and Loads

Bus Speed	***Connectors (Loads)***
66 MHz	4
100 MHz	2
133 MHz	1

A PCI-X split transaction consists of two parts: Split Request and Split Completion. Split transactions may be used in conjunction with the following PCI-X commands (BCM57XX family supported):

- Memory Read Block
- Memory Read DWORD
- Interrupt Acknowledge
- Configuration Read
- Configuration Write

There are two players in Split Transactions: Requester and Completer. The requester is typically a PCI-X initiator (i.e., BCM57XX family) and the completer is the Target device. The BCM57XX family can handle one outstanding split memory read at any time. Two PCI transactions may be outstanding at any point—one memory read and one memory write transaction. However, the PCI interface does not support a split memory write transaction (not defined). Figure 41 on page 84 shows a split memory read timeline.

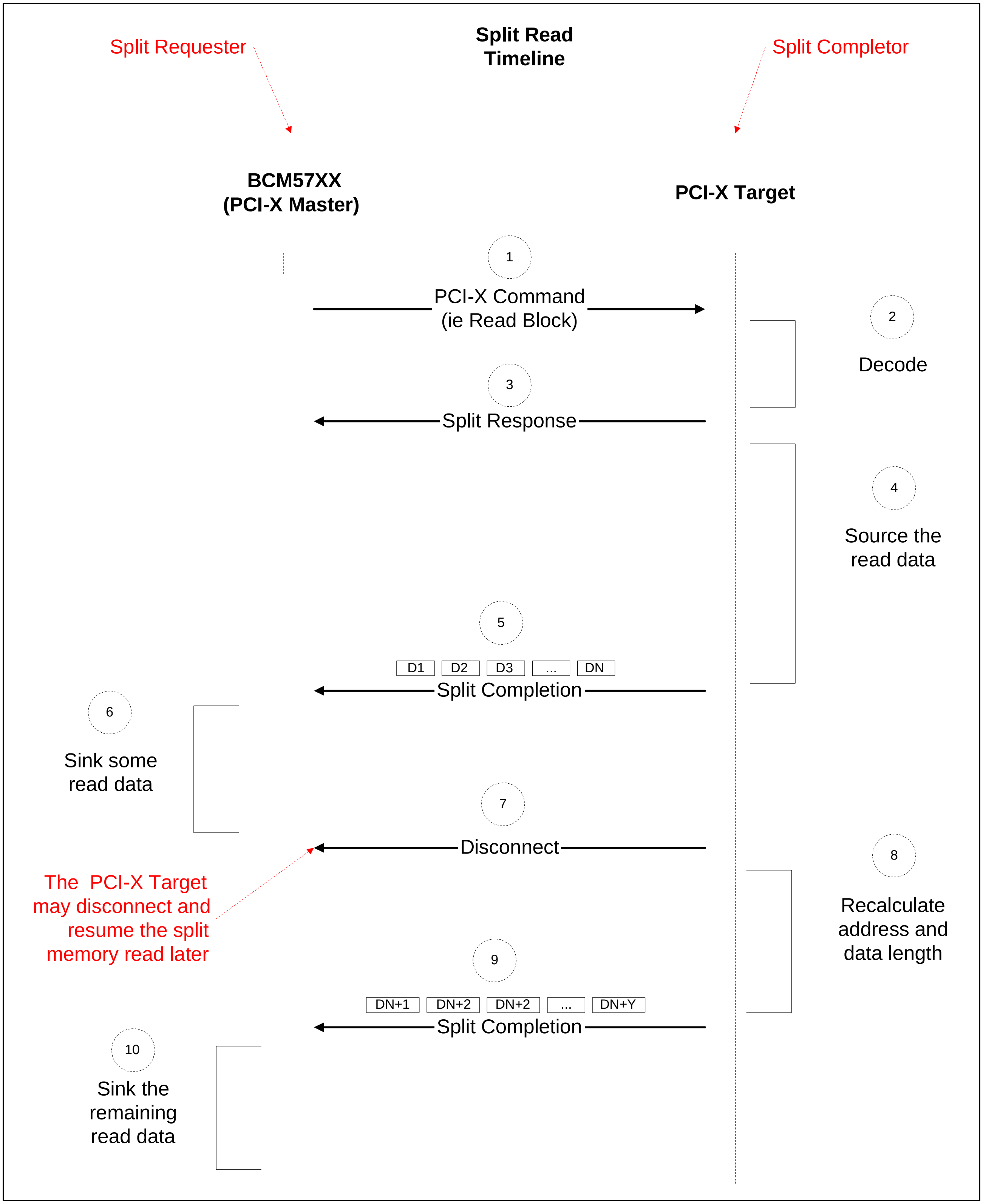


Figure 41: Split Memory Real Timeline

The phases of this split memory read example are as follows:

1. The BCM57XX family is the transaction initiator of a Split Memory Block Read command. The device is the Split transaction Requestor in this diagram.
2. The target device decodes the command and physical address. The target will claim this transaction by asserting DEVSEL and TRDY.
3. The target will issue a Split Response to defer the memory read.
4. The target will fetch the request memory from its local storage.
5. The target will issue a Split Completion transaction. The amount of bytes read and address will be included in the transaction message.
6. The BCM57XX family will sink the Split Completion data burst.
7. The BCM57XX family may disconnect after some amount of time. The target must still complete the Split Memory Read.
8. The target must recalculate the number of bytes and physical address. The target does not need to transfer the same data contents again, and waste bandwidth.
9. The target issues another Split Completion data burst, with the remaining data sourced.
10. The BCM57XX family completes the Split Completion transaction normally. The Split Memory Block Read is completed—this transaction is no longer outstanding.

The Broadcom PCI-X interface does not initiate split transactions; the BCM57XX family will respond with a Single Data Disconnect when another master attempts to initiate a block/read write. The BCM57XX family does not accept burst read/write transactions as a target device. However, the device will complete split transactions. The following table shows the PCI-X commands supported by the BCM57XX family.

Table 25: PCI-X Commands Supported by the* BCM57XX *Family

Command Encoding	*Description*	*Length*	*Usage*
0110	Memory Read DWORD	DWORD	BCM57XX family as PCI-X master or target[a]
0111	Memory Write	Burst	BCM57XX family as PCI-X master or target
1010	Configuration Read	DWORD	BCM57XX family as PCI-X target
1011	Configuration Write	DWORD	BCM57XX family as PCI-X target
1100	Split Completion	Burst	BCM57XX family as PCI-X target
1101	Dual Address Cycle	N/A	BCM57XX family as PCI-X master or target
1110	Memory Read Block	Burst	BCM57XX family as PCI-X master or target[a]
1111	Memory Write Block	Burst	BCM57XX family as PCI-X master or target[a]

[a] When using the BCM57XX family as a PCI-X target, break after one data cycle (i.e., for register space, a maximum of 32 bits; for memory and mailbox space, a maximum of 64 bits).

Initialization and Reset

See "Initialization and Reset" on page 720 for information on BCM57XX initialization and reset.

TERMINATION OF PCSXCAP SIGNAL

The termination used on the PCSXCAP signal determines the PCI-X operation of the BCM57XX family (see Figure 42). The BCM57XX reference NIC routes the PCIXCAP connector to a 0.01-µF capacitor; this configuration indicates PCI-X 133 MHz. Refer to section 9.10 of the v1.0 PCI-X specification for details on PCIXCAP. PCIXCAP is not routed to the BCM57XX packaging—discrete AC logic is required on the NIC board layout. PXICAP can also be tied to ground with a 10K resistor in parallel to a 0.01-µF cap. This configuration indicates PCI-X 66 MHz. The PCI bridge/chipset will sample PCIXCAP to determine the bus speed and capabilities of the device. The chipset uses a three-level comparator to determine the lowest common denominator—a common bus frequency.

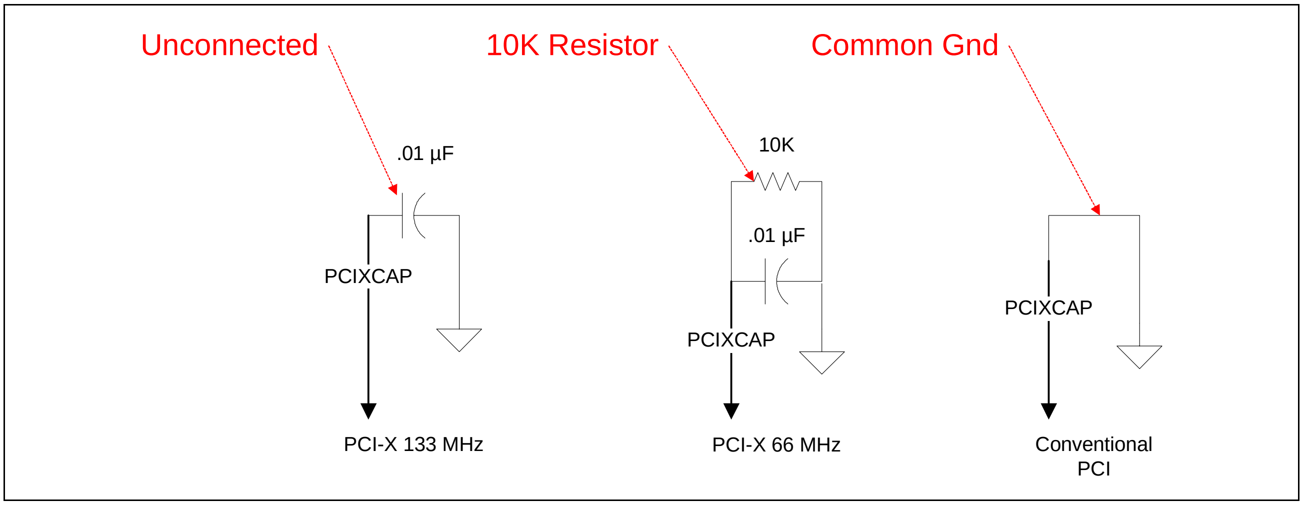


Figure 42: Terminating PCIXCAP to Determine PCI-X Operation

SELF-TEST

BIST

The BCM57XX NetXtreme family supports a manually controlled BIST function for use in manufacturing defect detection. This test is not intended for operation once the chip has been assembled into a NIC application. The BIST operation is controlled by the Enable BIST bit in the PCI Clock Control register (see "PCI Clock Control Register (Offset 0x74)" on page 334). The BIST operates only under specific chip conditions. The PCI BIST register (see "BIST Register (Offset 0x0F)" on page 306) cannot be written by host software—this register is read-only.

JTAG

IEEE 1149 compliant boundary scan JTAG is supported by the BCM57XX NetXtreme family. The IDCODE, BYPASS, EXTEST, and SAMPLE instructions of the IEEE standard are implemented. These instructions allow each pin on the part to be controlled and monitored from the JTAG serial interface pins. This industry standard technique allows the connections between the component die and the circuit board to be tested once the assembly has been built. The standard packaging for the BCM57XX NetXtreme family does not provide pinout for JTAG. An industry standard BSDL definition of the JTAG implementation is available from Broadcom Technical Support.

Section 4: NVRAM Configuration

Broadcom NetXtreme controllers require the use of an external non-volatile memory (NVRAM) device (Flash or SEEPROM), which contains a bootcode program that the controller's on-chip CPU core loads and executes upon release from reset. This external NVRAM device also contains many configuration items that direct the behavior of the controller, enable/disable various management and/or value-add firmware components, etc. All configuration settings are default-configured in the official release binary image files provided in Broadcom's CD software releases. However, the settings chosen as default by Broadcom may not be what best suits a particular OEM's application so may need to be changed by the OEM.

Details relating to the NVRAM can be found in *NetXtreme/NetLink NVRAM Access* Broadcom application note (Netxtreme-AN500-R). Some of the topics addressed by this application note include the following:

- Programming NVRAM (sample C code, x86 assembly)
- NVRAM map
- Configuration settings
- Bootcode
- Multiple boot agent (MBA), PXE, etc.
- Management firmware (ASF)

Note: NVRAM CRC-32: There are multiple distinct regions contained within the NVRAM map. Each of these regions has its own CRC-32 checksum value associated with it. If any data element contained within a region is modified, then that region's CRC-32 value must also be updated. Details relating to calculating the CRC-32 can be found in *Calculating CRC32 Checksums for Broadcom NetLink, NetXtreme, and NetXtreme II Controllers* Broadcom application note (NetXtreme_NetXtremeII-AN200-R).

Section 5: Common Data Structures

Theory of Operation

Several device data structures are common to the receive, transmit, and interrupt processing routines. These data structures are hardware-related and are used by device drivers to read and update state information.

Descriptor Rings

In order to send and receive packets, the host and the controller use a series of shared buffer descriptor rings to communicate information back and forth. Each ring is composed of an array of buffer descriptors that reside in host memory. These buffer descriptors point to either send or receive packet data buffers. The largest amount of data that a single buffer may contain is 65535 (64K-1) bytes (The length field in BD is 16 bits). Multiple descriptors can be used per packet in order to achieve scatter-gather DMA capabilities.

There are three main types of descriptor rings:

- Send Rings (Host Based or Controller Based)
- Receive Producer Rings
- Receive Return Rings

BCM5700 with external memory supports up to 16 Host Based or Controller Based Send Rings, Mini Receive Producer Ring, Standard Receive Producer Ring, Jumbo Receive Producer Ring, and 16 Receive Return Rings.

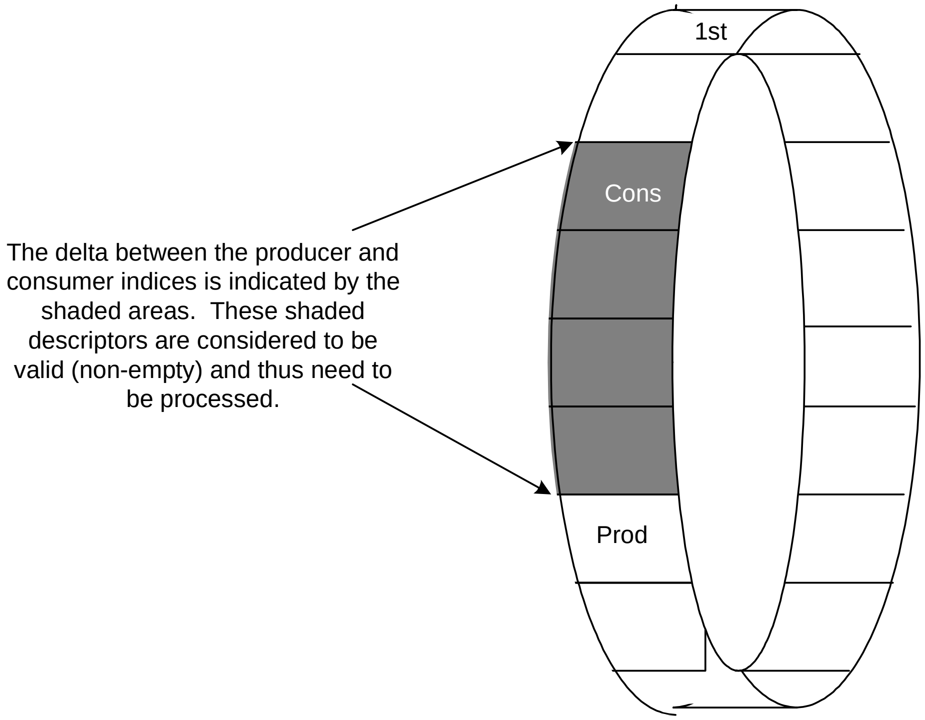
BCM5700 without external memory, BCM 5701, BCM5702, BCM5703C, BCM5703S, BCM5704C, and BCM5704S MACs support up to 4 Host Based or Controller Based Send Rings, Standard Receive Producer Ring, Jumbo Receive Producer Ring, and 16 Receive Return Rings.

BCM5705, BCM5788, BCM5721, BCM5751, BCM5714C, BCM5714S, BCM5715C, BCM5715S, and BCM5752 MACs support One Host Based Send Ring, One Receive Producer Ring, and One Receive Return Ring.

Producer and Consumer Indices

The Producer Index and the Consumer Index control which descriptors are valid for a given ring. Each ring will have its own separate Producer and Consumer Indices. When incremented, the Producer Index can be used to add elements to the ring. Conversely, when incremented, the Consumer Index is used to remove elements from the ring. The difference between the Producer and Consumer Indices mark which descriptors are currently valid in the ring (see Figure 43). When the Producer and Consumer Index are equal, the ring is empty. When the producer is one behind the consumer, the ring is considered to be full.

The drawing shows a generic host descriptor ring (could be either a send ring or a receive ring), and demonstrates how the consumer and producer indices are used to determine which descriptors in the ring are valid at any given moment in time.



The delta between the producer and consumer indices is indicated by the shaded areas. These shaded descriptors are considered to be valid (non-empty) and thus need to be processed.

Figure 43: Generic Ring Diagram

RING CONTROL BLOCKS

Each ring (send or receive) has a Ring Control Block (RCB) associated with it. Each RCB has the format shown in Table 26.

Table 26: Ring Control Block Format

Offset (bytes)	*31*	*16*	*15*	*0*
0x00	Host Ring Address			
0x04				
0x08	Max_len		Flags	
0x0c	NIC Ring Address (reserved in BCM5705)			

The fields are defined as follows:

- The Host Ring Address field contains the 64-bit host address of the first element in the ring. Basically, this field tells the controller precisely where in host memory the ring is located. This field only applies to rings that are located in host memory (i.e., for on-chip Send Rings this field is ignored). The Host_Ring_Address field contains the 64-bit address, in big-endian ordering, of the first Send BD in host memory. In NIC-based send rings, the field is ignored.
- The Flags field contains bits flags that contain control information about a given ring. The following table shows the two flags that are defined.

Table 27: Flag Fields for a Ring

Bits	*Name*	*Description*
0	RCB_FLAG_USE_EXT_RECV_BD	This bit should be used only for Receive Jumbo Rings. This bit indicates that the ring will use extended receive buffer descriptors (see "Receive Buffer Descriptors" on page 100).
1	RCB_FLAG_RING_DISABLED	Indicates that the ring is not in use.
15-2	Reserved	Reserved for future use. Should be set to 0.

- The Max_len field has a different meaning for different types of rings.
  - For Send rings and Receive Return rings, this field indicates the number elements in the ring.
  - For Host-Based Send Rings, the size of the ring is fixed at 512 entries and thus this field should be set to 512.
  - For Controller-Based Send Rings, this field can be 128 or 512 depending on number of Send Rings configured. The device internal Send BD memory can be used either as one 512 BD ring or four 128 BD rings. To configure one NIC based Send Ring to use 512 BD entries of internal memory, set Mode_Control.4x_SIZE_NIC_Based_Send_Rings (Bit 29 of the Mode Control register, see "Mode Control Register (Offset 0x6800)" on page 502) to 1. In order to configure four Send Rings, each with 128 BDs, set Mode_Control.4x_SIZE_NIC_Based_Send_Rings (Bit 29 of the Mode Control register, see "Mode Control Register (Offset 0x6800)" on page 502) to 0. When external memory is used with BCM5700 it is possible to configure up to 16 Send Rings, each with 512 BDs.
  - For Receive Return Rings, the only valid values are 1024 and 2048. It is very important to make sure that this value is set to 2048 when Receive Mini Producer Ring is enabled. This is because the Receive Return Ring size should be larger than the sum of sizes of Receive Producer Rings that are enabled.

Note: Configure the Receive Return Ring Max_Len = 2048 when the Receive Mini Producer Ring is enabled. The Receive Mini Ring is supported only with BCM5700 when external memory is used.

- For Receive Producer Rings, this field corresponds to the maximum size buffer associated with the elements in the ring. The received packets whose length is less than or equal to the Max_Len field of Receive Mini Producer Ring will consume a BD from Receive Mini Producer Ring. The received packets whose length is greater than the Max_Len field of Receive Mini Producer Ring but less than or equal to the Max_Len field of Receive Standard Producer Ring will consume a BD from Receive Standard Producer Ring. And the received packets whose length is greater than the Max_Len field of Receive Standard Producer Ring will consume a BD from Receive Jumbo Producer Ring. This Max_Len field is ignored in the Receive Jumbo Producer Ring. The BCM5705, BCM5788, BCM5721, BCM5751, BCM5714C, BCM5714S, BCM5715C, BCM5715S, and BCM5752 MAC devices support only One Receive Producer Ring. So for these controllers, this Max_len field of the RCB indicates the number of elements in the ring. The maximum programmable value of Max_len field is 512 for these controllers.

- The NIC Ring Address field contains the address where the BD cache is located in the internal NIC address space. This address is only valid for Receive Producer Rings. The Send Rings and Receive Return Rings do not require this field to be populated. The location within the NIC address map for Receive Producer Ring may be located in the Memory Maps and Pool Configuration section (see "Memory Maps and Pool Configuration" on page 171).

SEND RINGS

The BCM5700 with external memory supports up to 16 Host Based or Controller Based Send Rings. The BCM5700 without external memory, BCM5701, BCM5702, BCM5703C, BCM5703S, BCM5704C, and BCM5704S MACs support up to 4 Host Based or Controller Based Send Rings. Multiple Send Rings can be utilized by host software to select varying levels of priority and thus support varying levels of quality of service.

Table 28: Send Rings In BCM5700/5701/5702/5703C/5703S/5704C/5704S Devices

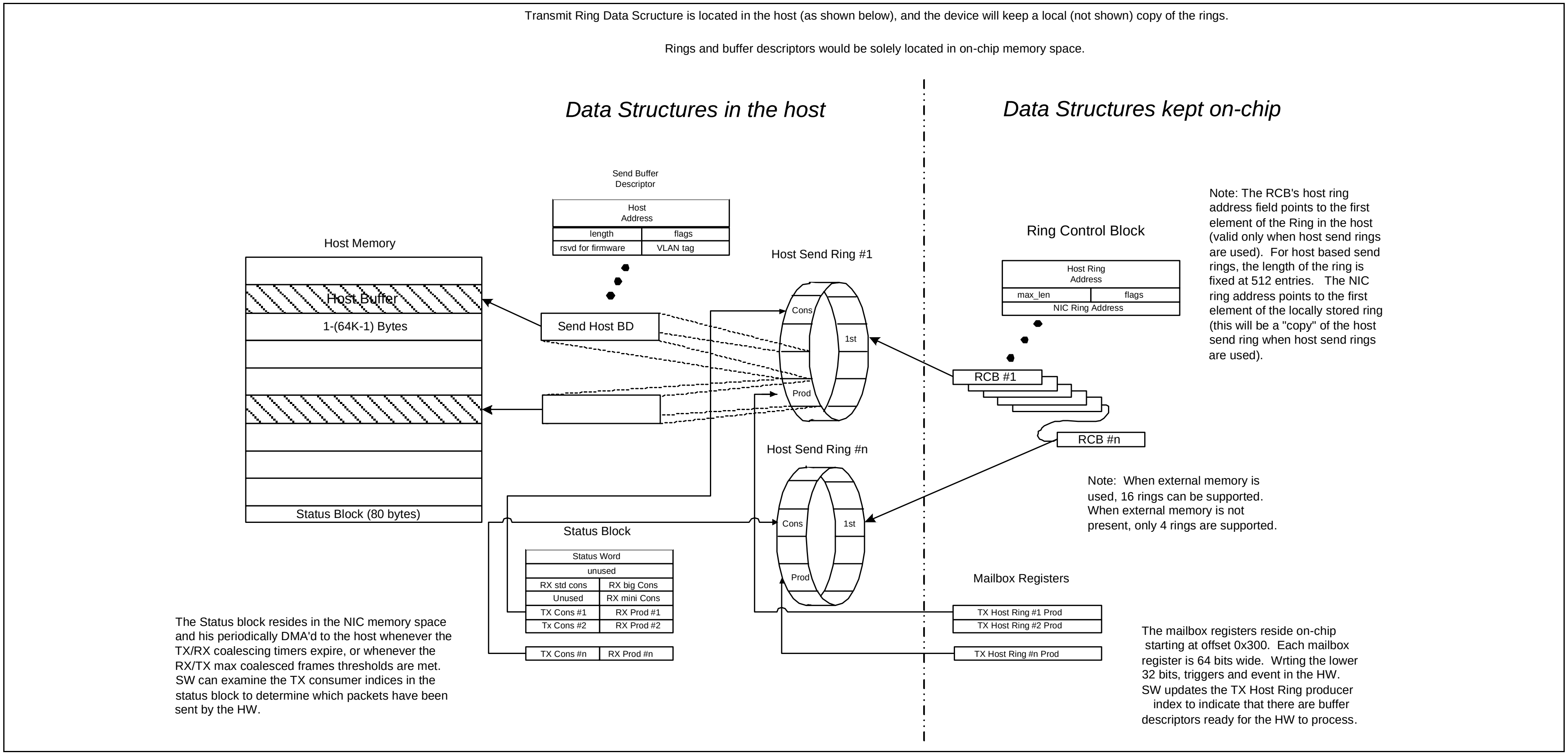
Description	*Internal Memory Only*	*External Memory*[a]	*Mode*
Number of Rings	4	16	
Buffer Descriptor Size(Bytes)	16	16	
Host Ring Size	512	512	Host-Based Rings
(# of Buffer Descriptors)	0	0	NIC-Based Rings
NIC Ring Size	128	128	Host-Based Rings
(# of Buffer Descriptors)	128	128	4X Ring Size = 0 and NIC-Based Rings
	512 (Only One Send Ring is Supported in this Configuration)	512	4X Ring Size = 1 and NIC-Based Rings

[a] Only the BCM5700 MAC supports external SSRAM memory.

The BCM5705, BCM5788, BCM5721, BCM5751, BCM5714C, BCM5714S, BCM5715C, BCM5715S, and BCM5752 MACs support only One Host Based Send Ring.

Normally The Host Driver Software uses Host Based Send Rings. A copy of these Host Send Rings will then be cached in the Controller. It is possible for host software to bypass Host-Based Send Rings and directly manipulate the on-chip Controller Based Send Rings. However, this is normally not advisable because this mechanism would require more PCI slave accesses to the controller, and that would put an unnecessary burden on the host CPU.

The Send Ring Producer Index is incremented by host software to add descriptors to the Send Ring (see Figure 44). By adding descriptors to the ring, the device is instructed to transmit packets that are composed of the buffers pointed to by the descriptors. A single transmit packet may be composed of multiple buffers that are pointed to by multiple send descriptors. There is no limit as to how many descriptors can be used per packet other than the limit on the number of descriptors in the ring itself.



Transmit Ring Data Scructure is located in the host (as shown below), and the device will keep a local (not shown) copy of the rings.

Rings and buffer descriptors would be solely located in on-chip memory space.

Note: The RCB's host ring address field points to the first element of the Ring in the host (valid only when host send rings are used). For host based send rings, the length of the ring is fixed at 512 entries. The NIC ring address points to the first element of the locally stored ring (this will be a "copy" of the host send ring when host send rings are used).

Note: When external memory is used, 16 rings can be supported. When external memory is not present, only 4 rings are supported.

The Status block resides in the NIC memory space and his periodically DMA'd to the host whenever the TX/RX coalescing timers expire, or whenever the RX/TX max coalesced frames thresholds are met. SW can examine the TX consumer indices in the status block to determine which packets have been sent by the HW.

The mailbox registers reside on-chip starting at offset 0x300. Each mailbox register is 64 bits wide. Wrting the lower 32 bits, triggers and event in the HW. SW updates the TX Host Ring producer index to indicate that there are buffer descriptors ready for the HW to process.

Figure 44: Transmit Ring Data Structure Architecture Diagram

Send Buffer Descriptors

The format of an individual Send Buffer Descriptor is shown in Table 29.

Table 29: Send Buffer Descriptors Format

Offset (bytes)	31 16	15 0
0x00	Host Address	
0x04		
0x08	Length	Flags
0x0c	Reserved	VLAN Tag

The fields are defined as follows:

- The Host Address field contains the 64-bit host address of the buffer that the descriptor points to. A length of 0 indicates that the descriptor does not have a buffer associated with it.
- The Flags field contains bits flags that contain control information for the device for transmitting the packets. The defined flags are listed in Table 30.

Table 30: Defined Flags for Send Buffer Descriptors

Bits	Name	Description
0	TCP_UDP_CKSUM[a]	If set to 1, the controller replaces the TCP/UDP checksum field of TCP/UDP packets with the hardware calculated TCP/UDP checksum for the packet associated with this descriptor.
1	IP_CKSUM	If set to 1, the controller replaces the IP checksum field of TCP/UDP packets with the hardware calculated IP checksum for the packet associated with this descriptor. This bit should only be set in the descriptor that points to the buffer that contains the IP header. It is assumed that the IP header is contained in a single buffer.
2	PACKET_END	If set to 1, the packet ends with the data in the buffer pointed to by this descriptor.
3	IP_FRAG[a]	If set to 1, it indicates to the controller that this BD is part of a frame that is fragmented at the IP layer. If multiple send rings are enabled, hardware completes the processing of all BDs in the IP fragment chain, before moving to the next enabled send ring.
4	IP_FRAG_END	If set to 1, it indicates to the controller that this BD is the last fragment in the last packet of a train of IP fragmented packets. If multiple send rings are enabled, hardware completes processing all BDs in the IP fragment chain, before moving to the next enabled send ring.
5	Reserved	Reserved for future use. Should be set to 0.
6	VLAN_TAG[a]	If set to 1, the device inserts an 802.1Q VLAN tag into the packet. The 16-bit TCI (Tag Control Information) field of four byte VLAN tag comes from the VLAN Tag field in the descriptor.
7	COAL_NOW	If set to 1, the device immediately updates the Send Consumer Index after the buffer associated with this descriptor has been transferred via DMA to NIC memory from host memory. An interrupt may or may not be generated according to the state of the interrupt avoidance mechanisms. If this bit is set to 0, then the Consumer Index is only updated as soon as one of the host interrupt coalescing conditions has been met.
8	CPU_PRE_DMA	If set to 1, the controller's internal CPU is required to act upon the packet before the packet is given to the internal *Send Data Initiator* state machine. Normally this bit should be set to 0.

Table 30: Defined Flags for Send Buffer Descriptors (Cont.)

Bits	Name	Description
9	CPU_POST_DMA[a]	If set to 1, the controller's internal CPU is required to act upon the packet before the packet is given to the internal *Send Data Completion* state machine. Normally this bit should be set to 0.
11-10	Reserved	Reserved for future use. Should be set to 0.
12	INSERT_SRC_ADDR[a]	If set to 1, the controller should insert a source address into the packet's Ethernet header. Normally, this bit is not set because host software already has built the packet to include the Ethernet source address.
14-13	CHOOSE_SRC_ADDR[a]	This two-bit field indicates which of the four MAC addresses should be inserted as the source address for the packet. These bits are meaningful only when bit 12 is set.
15	DON'T_GEN_CRC[a]	If set to 1, the controller will not append an Ethernet CRC to the end of the frame.

[a] Indicates that this bit should be set in all descriptors for a given packet if the desired capability is to be enabled for that packet.

Note: The UDP checksum engine does not span IP fragmented frames. The IP_FRAG and IP_FRAG_END flags do not enable UDP checksum capability, when the IP layer has fragmented the UDP message.

- The Length field specifies the length of the data buffer. The lengths for the buffers associated with a given packet will add up to the length of the packet.

Note: The 57XX family does not validate the value of the Length field and may generate an error on the PCI bus if the Length field has a value of 0. The host driver must ensure that the Length field is nonzero before enqueueing the BD onto the Send Ring.

- The VLAN Tag field is only valid when the VLAN_TAG bit of Flags field is set. This VLAN Tag field contains the 16-bit VLAN tag that is to be inserted into an 802.1Q (and 802.3ac) compliant packet by the controller. If VLAN tag insertion is desired, this field (and the flag) should be set in the first descriptor for that packet (i.e., the descriptor that points to the buffer that contains the Ethernet header).
- The Reserved field of Send BD provides a placeholder for future functionality. It could also be used by specialized firmware.

RECEIVE RINGS

The BCM57XX NetXtreme family of MACs support two types of Receive Descriptor Rings: Producer Rings and Return Rings (see Figure 45). Descriptors in the Producer Rings point to free buffers in the host. When the controller receives a packet and consumes a receive buffer, the controller will modify and writeback the descriptor for the consumed buffer into the given Receive Return Ring. Basically the Producer Rings contain descriptors that point to buffers that the controller is free to use, whereas the Return Rings contain descriptors that the device has used and await processing from host software.

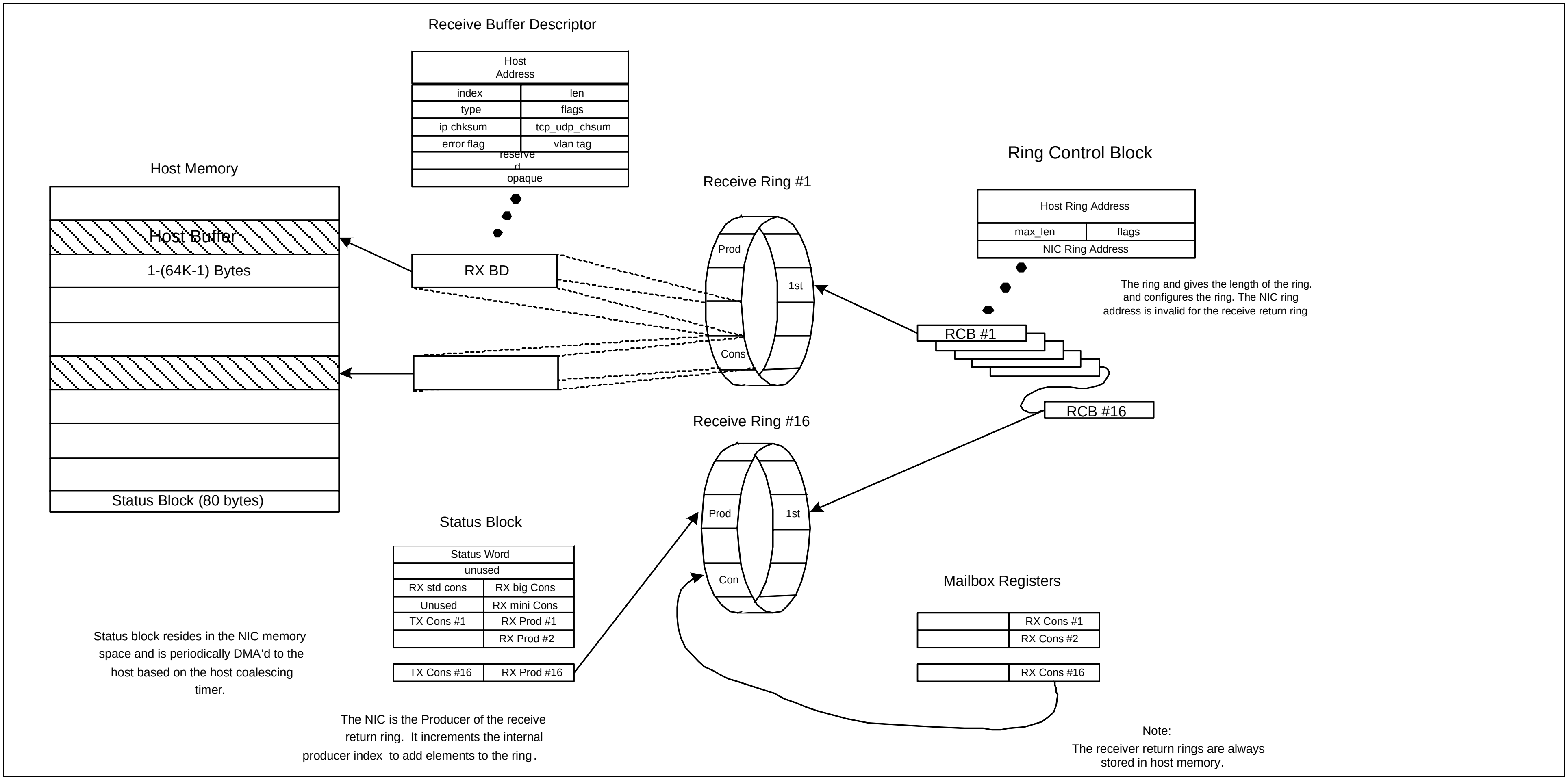

Receive Buffer Descriptor
Host Address
index
len
type
flags
ip chksum
tcp_udp_chsum
error flag
vlan tag
reserved
opaque
Host Memory
Host Buffer
1-(64K-1) Bytes
Status Block (80 bytes)
RX BD
Receive Ring #1
Prod
1st
Cons
Ring Control Block
Host Ring Address
max_len
flags
NIC Ring Address
The ring and gives the length of the ring. and configures the ring. The NIC ring address is invalid for the receive return ring
RCB #1
RCB #16
Receive Ring #16
Prod
1st
Con
Status Block
Status Word
unused
RX std cons
RX big Cons
Unused
RX mini Cons
TX Cons #1
RX Prod #1
RX Prod #2
TX Cons #16
RX Prod #16
Status block resides in the NIC memory space and is periodically DMA'd to the host based on the host coalescing timer.
The NIC is the Producer of the receive return ring. It increments the internal producer index to add elements to the ring.
Mailbox Registers
RX Cons #1
RX Cons #2
RX Cons #16
Note:
The receiver return rings are always stored in host memory.

Figure 45: Receive Return Ring Memory Architecture Diagram

Receive Producer Rings

There are three types of producer rings: the Mini Receive Ring (Table 31), the Standard Receive Ring (Table 32), and the Jumbo Receive Ring (Table 33). The Mini Receive Ring and the Jumbo Receive Ring are optional. The producer rings reside in the host and point to empty host receive buffers that will later be filled with received packet data. The controller will internally cache a copy of the three rings. The three producer rings differ from each other in the size of receive buffers that they point to. The Mini Receive Ring is intended to point to small buffers that can be used to store very small packets. The Standard Receive Ring is intended to point to buffers that are large enough to store a standard-sized Ethernet packet (1522 Byte VLAN Tagged Packets and/or 1518 Byte Untagged Packets). The Jumbo Receive Ring is intended to point to very large buffers that can store packets larger than 1522 Byte Standard Ethernet Packets (e.g., 9K Jumbo Frames). The rationale behind the three producer rings is the efficient host memory utilization by using small buffers for smaller packets and large buffers for large packets.

When the host software driver has a free host receive packet buffer available for incoming packets, it will fill out a receive producer descriptor of appropriate producer ring depending on buffer size and have the descriptor point to the available buffer. Host software will then update the producer index for that receive producer ring to indicate to the controller that there is a newly available receive buffer. After the controller fetches and caches (e.g., consumes) this receive producer descriptor, the controller will update the consumer index for the appropriate receive producer ring.

The external SSRAM memory has no effect on the size of the buffer descriptor. The Standard Producer Ring and Jumbo Producer Ring (Table 33) may be enabled with or without external memory. The Mini Producer Ring (Table 31) can only be enabled when external SSRAM is present and hence is supported only on BCM5700 with external memory. There is an extended RX Buffer Descriptor that can be used only with Jumbo Producer Ring. The extended RX BD supports scatter gather, so the entire 9K of physical memory does not need to be mapped into one contiguous host memory location. The extended RX Producer BD is enabled by setting the RCB_FLAG_USE_EXT_RECV_BD flag in the flags field of the RCB. The extended RX BD has fields for four host addresses and four lengths.

The BCM5700 with External Memory Supports all three Receive Producer Rings (Mini Producer Ring, Standard Producer Ring, and Jumbo Producer Ring). The Jumbo Producer Ring may use either Standard Rx BDs or Extended Rx BDs.

The BCM5700 without External Memory, BCM5701, BCM5702, BCM5703C, BCM5703S, BCM5704C, and BCM5704S devices support Standard and Jumbo Producer Rings. The Jumbo Producer Ring may use either Standard Rx BDs or Extended Rx BDs.

The BCM5705, BCM5788, BCM5721, BCM5751, BCM5714C, BCM5714S, BCM5715C, BCM5715S, and BCM5752 devices support only Standard Producer Ring. The BCM5714C, BCM5714S, BCM5715C, and BCM5715S devices support a configuration option of using either the Standard Rx BDs or Extended Rx BDs.

Table 31: Mini Receive Producer Ring(BCM5700 MAC Only)

Description	External Memory Mode[a]
Number of Rings	1
Buffer Descriptor Size(Bytes)	32
Host Ring Size (# of Buffer Descriptors)	1024
NIC Cache Size (# of Buffer Descriptors)	256

[a] Only the BCM5700 MAC supports external SSRAM memory so Mini Ring is supported only on BCM5700.

Table 32: Standard Receive Producer Ring

Description	*Internal Memory Mode*	*External Memory Mode*[a]
Number of Rings	1	1
Buffer Descriptor Size(Bytes)	32	32
Host Ring Size (# of Buffer Descriptors)	512	512
NIC Cache Size (# of Buffer Descriptors)	128	128

[a] Only the BCM5700 MAC supports external SSRAM memory.

Table 33: Jumbo Receive Producer Ring (BCM5700/5701/5702/5703C/5703S/5704C/5704S Only)

Description	*Internal Memory Mode*	*External Memory Mode*[a]	*Mode*
Number of Rings	1	1	
Buffer Descriptor Size(Bytes)	32	32	Extended BD's = 0
	64	64	Extended BD's = 1
Host Ring Size (# of Buffer Descriptors)	256	256	
NIC Cache Size (# of Buffer Descriptors)	If Standard RX BD is used, there are 128 entries in the NIC ring. If Extended RX BD is used, there are only 64 entries in the NIC ring.	If Standard RX BD is used, there are 128 entries in the NIC ring. If Extended RX BD is used, there are only 64 entries in the NIC ring.	

[a] Only the BCM5700 MAC supports external SSRAM memory.

Receive Return Rings

The BCM57XX NetXtreme family supports up to sixteen Receive Return Rings. Multiple rings can be utilized by host software to select varying levels of priority and thus support varying levels of quality of service.

When the NIC receives a packet, it will DMA that packet to a host receive packet data buffer whose size is appropriate for the incoming packet (see Section 6: "Receive Data Flow" on page 116). Earlier the NIC will have received ownership of that data buffer via an update of the producer index for one of the three receive producer rings. After the controller does the packet data write DMA, it will DMA a corresponding buffer descriptor into the appropriate receive return ring. The buffer descriptor that is returned in the receive return ring will be slightly modified from the original buffer descriptor that the controller fetched out of the receive producer ring. After the controller has completed the DMA of the receive return ring descriptor, the controller will update its internal copy of the producer index for that particular receive return ring. That new value for that receive return ring producer index will be included in the next status block update that is made to the host. Host software will know that new packets have been received when host software sees that a receive return ring producer index has updated.

Table 34: Receive Return Rings

Description	*BCM5700 with External Memory*	*BCM5700 with Internal Memory/5701/5702/5703C/ 5703S/5704C/5704S Devices*	*BCM5705/5788/ 5721/5751/5752 Devices*	*BCM5714C/5714S/ 5715C/5715S Devices*
Number of Rings	16	16	1	1
Buffer Descriptor Size (Bytes)	32	32	32	32
Host Ring Size (# of Buffer Descriptors)	2048 if Mini Ring is Enabled 1024 if Mini Ring is Disabled	1024	512	512 if Standard Rx BDs are used. 256 if Extended Rx BDs are used
NIC Cache Size (# of Buffer Descriptors)	0	0	0	0

Receive Buffer Descriptors

The format of Standard Receive Buffer Descriptors (in both producer rings and return rings) is shown in Table 35. See "Extended Receive Buffer Descriptor" on page 122 for a description of the format of Extended Receive Buffer Descriptors that would be needed for support of Jumbo Frames.

Table 35: Receive Descriptors Format

Offset (bytes)	*31*	*16*	*15*	*0*
0x00 0x04	Host Address			
0x08	Index		Length	
0x0c	Type		Flags	
0x10	IP_Cksum		TCP_UDP_Cksum	
0x14	Error_Flags		VLAN tag	
0x18	Reserved			
0x1C	Opaque			

The fields are defined as follows:

- The Host Address field contains the 64-bit host address of the buffer that the descriptor points to. A length of 0 indicates that the descriptor does not have a buffer associated with it.
- The Length field specifies the length of the data buffer. For Producer Rings this value is set by the host software to correspond to the size of the buffer that is available for a receive packet. Once a packet has been received, the controller will modify this length field to correspond to the length of the packet received. A value of 0 indicates that there is no valid data in the buffer.
- The Index field is set by host software in the descriptors in the producer rings. When the controller uses a given buffer descriptor, it will opaquely pass the Index field for that buffer descriptor through to the corresponding descriptor in the return ring. This index field of the BD in Return Ring is then used by the host software to associate the BD in Return Ring with the BD in Producer Ring that points to the given receive buffer.
- The Flags field contains bits flags that contain control information about a given descriptor. The defined flags are listed in Table 36.

Table 36: Defined Flags for Receive Buffers

Bits	*Name*	*Description*
15	Reserved	Reserved for future use. Should be set to 0.
14	TCP_UDP_IS_TCP	In producer rings this bit should be set to 0. In return rings this bit will be set to 1 by the controller if the controller calculated that the incoming packet was a TCP packet. If the packet is UDP or a non IP frame, then this bit should be set to 0.
13	TCP_UDP_CHECKSUM	In producer rings this bit should be set to 0. In return rings this bit will be set to 1 by the controller if the controller calculated that the TCP or UDP checksum in the corresponding incoming packet was correct.
12	IP_CHECKSUM	In producer rings this bit should be set to 0. In return rings this bit will be set to 1 by the controller if the controller calculated that the IP checksum in the corresponding incoming packet was correct.
11	MINI_RING	The driver should set this to 1 for the descriptors in the Mini Receive Ring. The controller will not actually do anything differently based on this bit, rather it will just copy this bit through to the receive return ring. Thus, if the bit is set to 1 in the return ring, the host driver will know that the packet came from the Mini Receive Ring rather than the Standard Return Ring.

Table 36: Defined Flags for Receive Buffers (Cont.)

Bits	Name	Description
10	FRAME_HAS_ERROR	If set to 1 in a return ring, it indicates that the controller detected an error. The specific type of error is specified in the Error_Flag field of the receive return descriptor.
9-7	Reserved	Reserved for future use. Should be set to 0.
6	VLAN_TAG*	If set to 1 in a return ring, it indicates that the packet associated with this buffer contained a four-byte 802.1Q VLAN tag. The 16 VLAN ID is stripped from the packet and located in the VLAN tag field in the descriptor.
5	BD_FLAG_JUMBO_RING	Indicates that this packet came from the Jumbo Receive Ring, not the Standard Receive Ring (for receive BDs only). This must be set by the driver; it is just copied through opaquely by the NIC firmware.
4-3	Reserved	Reserved for future use. Should be set to 0.
2	PACKET_END	If set to 1, the packet ends with the data in the buffer pointed to by this descriptor.
1-0	Reserved	Reserved for future use. Should be set to 0.

- The Type field is used internally by the controller. In producer rings it should be set to 0, and in return rings it should be ignored by host software.
- The TCP_UDP_Cksum field holds the TCP/UDP checksum that the controller calculated for all data following the IP header given the length defined in the IP header. If the Receive No Pseudo-header Checksum bit is set (see "Mode Control Register (Offset 0x6800)" on page 502) to 1, then the pseudo-header checksum value is not added to this value. Otherwise, the TCP_UDP_Cksum field includes the pseudo-header in the controller's calculation of the TCP or UDP checksum. If the packet is not a TCP or UDP packet, this field has no meaning. Host software should zero this value in the producer ring descriptors. If the host is capable of TCP or UDP checksum off load, then host software may examine this field in the return rings to determine if the TCP or UDP checksum was correct.
- The IP_Cksum field holds the IP checksum that the controller calculated for the IP header of the received packet. If the packet is not an IP packet, this field has no meaning. Host software should zero this value in the producer ring descriptors. If the host is capable of IP checksum off load, then host software may examine this field in the return rings to determine if the IP was correct. A correct value would be 0 or 0xFFFF.
- The VLAN Tag field is only valid when the VLAN_TAG bit is set. This field contains the 16-bit VLAN ID that was extracted from an incoming packet that had an 802.1Q (and 802.3ac) compliant header.
- The Error_Flags field contains bits flags that contain error information about an incoming packet that the descriptor is associated with. The bits in this field are only valid if the FRAME_HAS_ERROR bit is set in the Flags field in the descriptor. The defined error flags are listed in Table 37.

Table 37: Defined Error Flags for Receive Buffers

Bits	*Name*	*Description*
31-9	Reserved	Reserved for future use. Should be set to 0.
8	GIANT_PKT_RCVD	If set to 1, the received packet was longer than the maximum packet length value set in the Receive MTU Size register (see "Receive MTU Size Register (Offset 0x43C)" on page 387). The data in the received packet was truncated at the length specified in the Receive MTU Size register.
7	TRUNC_NO_RES	If set to 1, the received packet was truncated because the controller did not have the resources to receive a packet of this length.
6	LEN_LESS_64	If set to 1, the received packet was less than the required 64 bytes in length.
5	MAC_ABORT	If set to 1, the MAC aborted due to an unspecified internal error while receiving this packet. The packet could be corrupted.
4	ODD_NIBBLE_RX_MII	If set to 1, the received packet contained an odd number of nibbles. Thus, packet data could be corrupt.
3	PHY_DECODE_ERR	If set to 1, while receiving this packet the device encountered an unspecified frame decoding error. This packet could be corrupted. This bit is set for valid packets that are received with a dribble nibble. True alignment errors will be dropped by that mac and never show up to the driver.
2	LINK_LOST	If set to 1, link was lost while receiving this frame. Therefore this packet is incomplete.
1	COLL_DETECT	If set to 1, a collision was encountered while receiving this packet.
0	BAD_CRC	If set to 1, the received packet has a bad Ethernet CRC.

- The reserved field provides a placeholder for future functionality. It could also be used by specialized firmware.
- The Opaque field is reserved for the host software driver. Any data placed in this field in a producer ring descriptor will be passed through unchanged to the corresponding return ring descriptor.

STATUS BLOCK

The Status Block is another shared memory data structure that is located in host memory. A copy of the Status Block is also kept in the controller (see "Status Block Base Address Register (Offset 0x3C44)" on page 457). The full Status Block is 80 bytes in length, when all Send and Receive rings are enabled. Host software will need to allocate up to 80 bytes of non-paged memory space for the full Status Block and set the Status Block Host Address register (see "Status Block Host Address Register (Offset 0x3C38)" on page 457) to point to the host memory physical address reserved for this structure. If host software enables a subset of the total available Send/Receive rings, the BCM57XX may be configured to update a partial status block; the unused portions of the status block are subsequently not DMA'd to host memory. Partial status block updates better utilize PCI bus bandwidth and decrease latency for other BCM57XX DMA transactions.

The controller will update the Status Block to host memory prior to a host coalescing interrupt or MSI. The frequency of these Status Block updates is determined by the host coalescing logic (see "Host Coalescing Engine" on page 62). There are many software configurable coalescing parameters that a device driver can manipulate to optimize the frequency of status block updates for a particular application or operating system.

The BCM57XX family supports partial status block updates to host memory (see Figure 46), and the host driver must set the Status Block Size bits in the Host Coalescing Mode register (see "Host Coalescing Mode Register (Offset 0x3C00)" on page 452).

Note: The BCM5700 MAC prior to the C0 revision does not support partial status block updates to host memory. (See "Revision Levels" on page 5.)

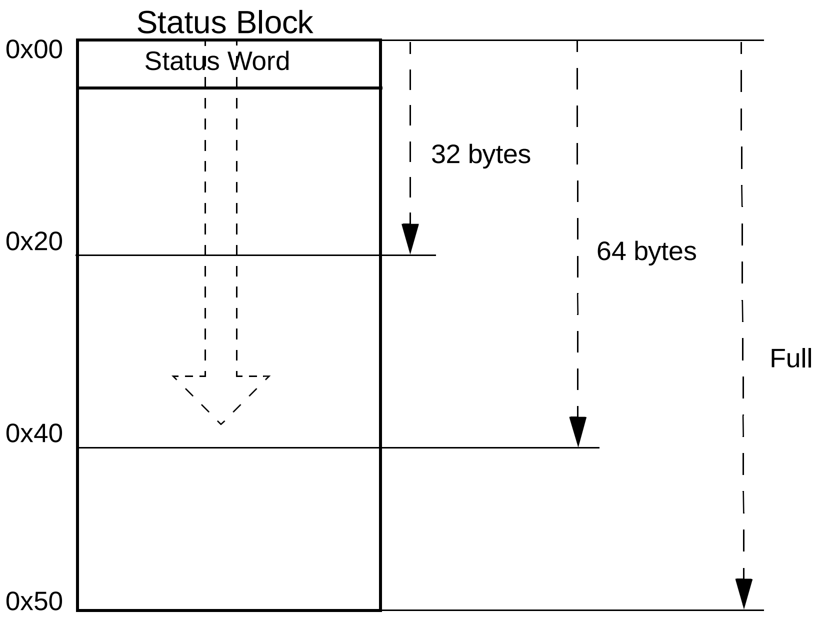


Figure 46: Partial Status Block Updates

The Status Block contains some of the Producer and Consumer indices for the rings described in "Descriptor Rings" on page 89. These Producer and Consumer indices allow host software to know what the current status of the controller is regarding its processing of the various send and receive rings. From information in the status block a software driver can determine:

- Whether the Status Block has been recently updated (via a bit in the status word).
- Whether the Link State has changed (via a bit in the status word).
- Whether the controller has recently received a packet and deposited that packet into host memory for a given ring (via the Receive Return Ring Producer Indices).
- Which host receive descriptors that controller has fetched, and it will consume when future packets are received (via the Receive Producer Ring Consumer Indices).
- Whether the controller has recently completed a transmit descriptor buffer DMA for a given ring (via the Send Ring Consumer Indices).

STATUS BLOCK FORMAT

The BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714C, BCM5714S, BCM5715C, and BCM5715S devices support only one Send Ring, one Receive Producer Ring, and one Receive Return Ring. So the Status Block is always 20 bytes long for these devices and its format is show below. Also the Status Block no longer exists in the controller memory space for these devices. The status block will still be DMAed to the host memory based on the host coalescing criteria. The host CPU can access the most current index through the register space.

Table 38: Status Block Format: BCM5705/5788/5721/5751/5752/5714/5715

Offset	***31***	***16***	***15***	***0***
0x00	Status Word			
0x04	Status Tag			
0x08	Receive Standard Consumer Index[a]		Unused	
0x0C	Unused			
0x10	Send BD Consumer Index[b]		Receive Return BD Producer Index[c]	

[a] The Receive Standard Consumer Index is also accessible at 0x3C54.

[b] The Send BD Consumer Index is also accessible at 0x3CC0.

[c] The Receive Return BD Producer Index is also accessible at 0x3C80.

The BCM5700, BCM5701, BCM5702, BCM5703C, BCM5703S, BCM5704C, and BCM5704S support multiple Send Rings and Receive Rings. For these devices, the full status block is 80 bytes long and its format is shown in Table 39.

Table 39: Status Block Format (all others)

Offset	***31***	***16***	***15***	***0***
0x00	Status Word			
0x04	Status Tag			
0x08	Receive Standard Consumer Index		Receive Jumbo Consumer Index	
0x0C	Reserved		Receive Mini Consumer Index	
0x10	Send Ring 1 Consumer Index		Receive Return Ring 1 Producer Index	
0x14	Send Ring 2 Consumer Index		Receive Return Ring 2 Producer Index	
0x18	Send Ring 3 Consumer Index		Receive Return Ring 3 Producer Index	
0x1C	Send Ring 4 Consumer Index		Receive Return Ring 4 Producer Index	
0x20	Send Ring 5 Consumer Index		Receive Return Ring 5 Producer Index	
0x24	Send Ring 6 Consumer Index		Receive Return Ring 6 Producer Index	
0x28	Send Ring 7 Consumer Index		Receive Return Ring 7 Producer Index	
0x2C	Send Ring 8 Consumer Index		Receive Return Ring 8 Producer Index	
0x30	Send Ring 9 Consumer Index		Receive Return Ring 9 Producer Index	
0x34	Send Ring 10 Consumer Index		Receive Return Ring 10 Producer Index	
0x38	Send Ring 11 Consumer Index		Receive Return Ring 11 Producer Index	
0x3C	Send Ring 12 Consumer Index		Receive Return Ring 12 Producer Index	
0x40	Send Ring 13 Consumer Index		Receive Return Ring 13 Producer Index	
0x44	Send Ring 14 Consumer Index		Receive Return Ring 14 Producer Index	
0x48	Send Ring 15 Consumer Index		Receive Return Ring 15 Producer Index	
0x4C	Send Ring 16 Consumer Index		Receive Return Ring 16 Producer Index	

The fields are defined as follows:

- The Status word field contains bit flags that contain error information about the status of the controller. The defined flags are listed in Table 40.

Table 40: Status Word Flags

Bit	Name	Description
0	Updated	This bit is always set to 1 each time the Status Block is updated in the host via DMA. It is expected that host software would clear this bit in the status block each time it examines the status block. This provides the host driver with a way of knowing whether the Status Block has been updated since the last time the driver looked at the Status Block.
1	Link State Changed	See "PHY Setup and Initialization" on page 250 for a description of PHY setup required when link state changes. This method of determining link change status provides a small performance increase over doing a PIO read of the Ethernet MAC Status register (see "Ethernet MAC Status Register (Offset 0x404)" on page 379). • BCM5700 MAC pre-C0 revision: If set to 1, the state of the Ethernet link has changed since the last status block update. This is really the Link State rather than a Link Changed bit. • BCM5700 MAC since C0 revision and the rest of the BCM57XX family: If set to 1, the interrupt was asserted due to a link change event.
2	Error	When this bit is asserted by the chip, the following conditions may have occurred. Bit 2 of the status word is the OR of: • All bits in Flow Attention register (0x3c48) (see "Flow Attention Register (Offset 0x3C48)" on page 457). • MAC_ATTN—Events from the MAC block (see "Ethernet MAC Status Register (Offset 0x404)" on page 379). • DMA_EVENT—Events from the following blocks: - MSI (see "MSI Status Register (Offset 0x6004)" on page 499). - DMA_RD (see "Read DMA Status Register (Offset 0x4804)" on page 479). - DMA_WR (see "Write DMA Status Register (Offset 0x4C04)" on page 482). • RXCP_ATTN—Events from RX RISC (see "RX RISC State Register (Offset 0x5004)" on page 485). • TXCP_ATTN—Events from TX RISC (see "TX RISC State Register (Offset 0x5404)" on page 488).

- The Status Tag field contains an unique eight-bit tag value in bits 7:0 when the Status Tagged Status Mode bit of the Miscellaneous Host Control register (see "Miscellaneous Host Control Register (Offset 0x68)" on page 325) is set to 1. This Status Tag can be returned to the Mailbox 0 register at location 31:24 (see "Interrupt Mailbox 0 Register (Offset 0x200)" on page 372) by host driver. When the remaining Mailbox 0 register bits 23:0 are written as 0, the tag field of the Mailbox 0 register is compared with the tag field of the last Status Block to be DMAed to host. If the tag returned is not equivalent to the tag of the last Status Block DMAed to the host, the interrupt state is entered.
- The Receive Jumbo Ring Consumer Index field contains the controller's current Consumer Index value for the Receive Producer Jumbo Ring. This field indicates how many receive descriptors are in the jumbo ring that the controller has consumed. For more information regarding this ring, see "Receive Producer Rings" on page 97.
- The Receive Standard Ring Consumer Index field contains the controller's current Consumer Index value for the Receive Producer Standard Ring. This field indicates how many receive descriptors are in the standard ring that the controller has consumed. For more information regarding this ring, see "Receive Producer Rings" on page 97.
- The Receive Mini Ring Consumer Index field contains the controller's current Consumer Index value for the Receive Producer Mini Ring. This field indicates how many receive descriptors are in the mini ring that the controller has consumed. For more information regarding this ring, see "Receive Producer Rings" on page 97.

- The Receive Return Rings 0-15 Producer Indices fields contain controller's current Producer Index value for the each of the Receive Return Rings. When the controller receives a packet and writes that packet data into host memory via DMA, it will increment the Producer Index for the corresponding Receive Return ring. When a Producer Index is incremented, it is a signal to software that a newly arrived receive packet is ready to be processed.
- The Send Rings 0-15 Consumer Indices fields contain controller's current Consumer Index value for the each of the Send Rings. When the controller completes the read DMA of the host buffer associated with a send BD, the controller will update the corresponding Send Ring Consumer Index. This provides the host software with an indication that the controller has buffered this send data and, therefore, the host software may free the buffer that was just consumed by the device.

STATISTICS BLOCK

Statistics are maintained by the NIC in Statistics Memory Block and transferred to host memory (see "Statistics Host Address Register (Offset 0x3C30)" on page 456) periodically based on the Statistics Ticks Counter register (see "Statistics Ticks Counter Register (Offset 0x3C28)" on page 456). All counters are cleared when a NIC reset occurs. All statistics are eight bytes long unless specified differently.

This statistics memory block does not exist in BCM5705, BCM5788, BCM5721, BCM5751, BCM5714C, BCM5714S, BCM5715C, BCM5715S, and BCM5752 MACs. Instead, these devices support the hardware statistics through "Statistics Registers" on page 407. In the device register space, these devices also support the statistics shown in Table 43 on page 109 through Table 49 on page 110, and Table 52 and Table 53 on page 113.

Note: For these controllers, the Statistics are not periodically DMAed to host memory.

MAC STATISTICS

MIB Statistics are based on RFC 1643. These statistics are generated by the TX MAC and the RX MAC (see Table 41).

Table 41: MAC Statistics

MIB Definition	*Description*
dot3StatsAlignmentErrors	A count of frames received on a particular interface that are not an integral number of octets in length and do not pass the FCS check.
dot3StatsFCSErrors	A count of frames received on a particular interface that are an integral number of octets in length and do not pass the FCS check.
dot3StatsSingleCollisionFrames	A count of successfully transmitted frames on a particular interface for which transmission is inhibited by exactly one collision.
dot3StatsMultipleCollisionFrames	A count of successfully transmitted frames on a particular interface for which transmission is inhibited by more than one collision.
dot3StatsDeferredTransmissions	A count of frames for which the first transmission attempt on a particular interface is delayed because the medium is busy.
dot3StatsLateCollisions	The number of times that a collision is detected on a particular interface later than 512 bit-times into the transmission of a packet.
dot3StatsExcessiveCollisions	A count of frames for which transmission on a particular interface fails due to excessive collisions.
dot3StatsInternalMacTransmitErrors	A count of frames for which transmission on a particular interface fails due to an internal MAC sublayer transmit error.

Table 41: MAC Statistics (Cont.)

MIB Definition	*Description*
dot3StatsCarrierSenseErrors	The number of times that the carrier sense condition was lost or never asserted when condition was lost or never asserted when attempting to transmit a frame on a particular interface.
dot3StatsFrameTooLongs	A count of frames received on a particular interface that exceed the maximum permitted frame size.

Interface Statistics

MIB Statistics are based on RFC 1213 and RFC 1573 interfaces group. These statistics are generated by the RX MAC and the TX MAC (see Table 42). Not all the statistics associated with the MIBs exist here. In particular, items that the host/driver would better handle, such as IFINDEX are left for the driver to set and report.

Table 42: Interface Statistics

MIB Definition	*Description*
ifInDiscards	The number of inbound packets which were chosen to be discarded even though no error has been detected to prevent their being deliverable to a higher-layer protocol.
ifInErrors	The number of inbound packets that contained errors prevent them from being deliverable to a higher-layer protocol. The ifInError counter increments if any of the below conditions occur: • Packet with FCS error. • Packet with alignment error. • Packet size greater than and equal to 2^16 bytes long (causing the counter to overflow). • Packet with extension error. • Packet size greater than preprogrammed MTU and keep_Oversize is not enabled in the Receive MAC Mode register (see "Receive MAC Mode Register (Offset 0x468)" on page 391). • Packet size less than 64 bytes and Accept_Runts is not enabled in the Receive MAC Mode register. • Packet with invalid 802.3 length field, if length checking is enabled in the Receive MAC Mode register. ***Note:*** A packet dropped due to: • An EMAC internal error (e.g., FIFO overflow) would not cause ifInError to increment. • Address filtering would not cause ifInError to increment.
ifInUnknownProtos	The number of packets received via the interface, which were discarded because of an unknown or unsupported protocol.
ifOutDiscards	The number of outbound packets which were chosen to be discarded even though no errors had been detected to prevent their being transmitted.
ifOutErrors	The number of outbound packets that could not be transmitted because of errors.
ifHCInOctets (8 bytes)	The number of octets received on the interface, including framing characters.
ifHCInUcastPkts (8 bytes)	The number of packets delivered by this sublayer to a higher (sub)layer, which were not addressed to a multicast or broadcast address at this sublayer.
ifHCInMulticastPkts (8 bytes)	The number of packets delivered by this sublayer to a higher (sub)layer, which were addressed to a multicast address at this sublayer.
ifHCInBroadcastPkts (8 bytes)	The number of packets delivered by this sublayer to a higher (sub)layer, which were addressed to a broadcast address at this sublayer.
ifHCOutOctets (8 bytes)	The number of octets transmitted out of the interface, including framing characters.

Table 42: Interface Statistics (Cont.)

MIB Definition	Description
ifHCOutUcastPkts (8 bytes)	The number of packets that higher-level protocols requested be transmitted and that were not addressed to a multicast or broadcast address at this sublayer, including those that were discarded or not sent.
ifHCOutMulticastPkts (8 bytes)	The number of packets that higher-level protocols requested be transmitted and that were addressed to a multicast address at this sublayer, including those that were discarded or not sent.
ifHCOutBroadcastPkts (8 bytes)	The number of packets that higher-level protocols requested be transmitted, and that were addressed to a broadcast address at this sublayer, including those that were discarded or not sent.

MIB Network Interface Card Statistics

Host Interrupts

The statistics shown in Table 43 are maintained by the send data initiator engine.

Table 43: Send Data Initiator Host Interrupts Statistics

Value Name	Description
nicRingSetSendProdIndex	Number of times the NIC has seen updates to any Send Producer Index.
nicRingStatusUpdate	Number of times the status block was updated.
nicInterrupts	Number of interrupts generated by NIC.
nicAvoidedInterrupts	Number of interrupts avoided by NIC.

NIC BD Coalescing Thresholds

The statistics shown in Table 44 are maintained by the send data initiator engine.

Table 44: Send Data Initiator NIC BD Coalescing Thresholds Statistics

Value Name	Description
nicSendThresholdHit	Number of times Send Max Coalesce Frames Threshold hit.

The statistics shown in Table 45 are maintained by the receive list placement engine.

Table 45: Receive List Placement NIC BD Coalescing Thresholds Statistics

Value Name	Description
nicRecvThresholdHit	Number of times Recv Max Coalesce Frames Threshold hit.

DMA Resources

These statistics are generated by FTQ for monitoring any state machine which adds DMA descriptors to the either DMA engine. These state machines include the Send BD Initiator, the Receive BD Initiator, the Send Data Initiator, the Receive Data and Receive BD Initiator, and the Host Coalescing state machines.

The nicDmaReadQueueFull statistics shown in Table 46 are updated by the Send Data Initiator state machine.

Table 46: Send Data Initiator DMA Resources Statistics

Value Name	Description
nicDmaReadQueueFull	Number of times DMA read queue was full.
nicDmaReadHighPrioQueueFull	Number of times DMA read high priority queue was full.

The nicDmaWriteQueueFull statistics shown in Table 47 are updated by the Receive List Placement state machine periodically.

Table 47: Receive List Placement DMA Resources Statistics

Value Name	Description
nicDmaWriteQueueFull	Number of times DMA write queue was full.
nicDmaWriteHighPrioQueueFull	Number of times DMA write high priority queue was full.

MAC Resources

The nicSendDataCompQueueFull statistics shown in Table 48 are updated by the Send Data Initiator state machine.

Table 48: Send Data Initiator MAC Resources Statistics

Value Name	Description
nicSendDataCompQueueFull	Number of times send data completion FTQ was full.

The nicNoMoreRXBDs statistics shown in Table 49 are updated by the Receive List Placement state machine periodically.

Table 49: Receive List Placement MAC Resources Statistics

Value Name	Description
nicNoMoreRXBDs	Number of times NIC ran out of the Receive Buffer Descriptors.

Internal MAC Receive Statistics

The statistics shown in Table 50 are maintained by the receive MAC.

Table 50: Internal MAC Receive Statistics

Counter	*Description*
dot3StatsAlignmentErrors	A count of frames received on a particular interface that are not an integral number of octets in length and do not pass the FCS check.
dot3StatsFCSErrors	A count of frames received on a particular interface that are an integral number of octets in length and do not pass the FCS check.
dot3StatsSingleCollisionFrames	A count of succesfully transmitted frames on a particular interface for which transmission is inhibited by exactly one collision.
dot3StatsMultipleCollisionFrames	A count of succesfully transmitted frames on a particular interface for which transmission is inhibited by more than one collision.
dot3StatsDeferredTransmssions	A count of frames for which the first transmission attempt on a particular interface is delayed because the medium is busy.
dot3StatsLateCollisions	The number of times that a collision is detected on a particular interface later than 512 bit-times into the transmission of a packet.
dot3StatsExcessiveCollisions	A count of frames for which transmission on a particular interface fails due to excessive collisions.
dot3StatsInternalMACTransmitErrors	A count of frames for which transmission on a particular interface fails due to an internal MAC sublayer transmit error.
dot3StatsCarrierSenseErrors	The number of times that the carrier sense condition was lost or never asserted when attempting to transmit a frame on a particular interface.
dot3StatsFrameTooLongs	A count of frames received on a particular interface that exceeds the maximum permitted frame size.
ifInDiscards	The number of inbound packets that were chosen to be discarded even though no error has been detected to prevent their being deliverable to a higher-layer protocol.
ifInErrors	The number of inbound packets that contained errors prevent them from being deliverable to a higher-layer protocol.
ifInUnknownProtos	The number of packets received via the interface which were discarded because of an unknown or unsupported protocol.
ifOutDiscards	The number of outbound packets which were chosen to be discarded even though no errors had been detected to prevent their being transmitted.
ifOutErrors	The number of outbound packets that could not be transmitted because of errors.
ifHCInOctets	The number of octets received on the interface, including framing characters.
ifHCInUnicastPkts	The number of packets, delivered by this sublayer to a higher (sub)layer, which were not addressed to a multicast or broadcast address at this sublayer.
ifHCInMulticastPkts	The number of packets, delivered by this sublayer to a higher (sub)layer, which were addressed to a multicast address at this sublayer.
ifHCInBroadcastPkts	The number of packets, delivered by this sublayer to a higher (sub)layer, which were addressed to a broadcast address at this sublayer.
ifHCOutOctets	The number of octets transmitted out of the interface, including framing characters.
ifHCOutUnicastPkts	The number of packets that higher-level protocols requested be transmitted, and that were not addressed to a multicast or broadcast address at this sublayer, including those that were discarded or not sent.

Table 50: Internal MAC Receive Statistics (Cont.)

Counter	Description
ifHCOutMulticastPkts	The number of packets that higher-level protocols requested be transmitted, and that were addressed to a multicast address at this sublayer, including those that were discarded or not sent.
ifHCOutBroadcastPkts	The number of packets that higher-level protocols requested be transmitted, and that were addressed to a broadcast address at this sublayer, including those that were discarded or not sent.
etherStatsFragments	A frame size that is less than 64 bytes with a bad FCS.
xonPauseFramesReceived	MAC control frames with pause command and length equal to zero.
xoffPauseFramesReceived	MAC control frames with pause command and length greater than zero.
macControlFramesReceived	MAC control frames with no pause command.
xoffStateEntered	Transmitting is disabled.
etherStatsJabbers	Frames exceed jabber time.
etherStatsUndersizePkts	Frames with size less than 64 bytes.
inRangeLengthError	Frames with length not equal to actual bytes received.
outRangeLengthError	Frames with type greater than 1522 or less than 1536.
etherStatsPkts64Octets	Frames size equal to 64 bytes.
etherStatsPkts65to127Octets	Frames size between 65 and 127 bytes inclusive.
etherStatsPkts128to255Octets	Frames size between 128 and 255 bytes inclusive.
etherStatsPkts256to511Octets	Frames size between 256 and 511bytes inclusive.
etherStatsPkts512to1023Octets	Frames size between 512 and 1023 bytes inclusive.
etherStatsPkts1024to1518Octets	Frames size between 1024 and 1518 bytes inclusive.
etherStatsPkts1519to2047Octets	Frames size between 1519 and 2047 bytes inclusive.
etherStatsPkts2048to4095Octets	Frames size between 2048 and 4095 bytes inclusive.
etherStatsPkts4096to8192Octets	Frames size between 4096 and 8192 bytes inclusive.
etherStatsPkts8192to9022Octets	Frames size between 8192 and 9022 bytes inclusive.

Internal MAC Transmit Statistics

The statistics shown in Table 51 are maintained by the transmit MAC.

Table 51: Internal MAC Transmit Statistics

Value Name	Description
etherStatsCollisions	Number of collisions experienced.
outXonSent	Sent Xon.
outXoffSent	Sent Xoff.
flowControlActive	Currently flow controlled.
dot3Collided2Times	Number of frames that experienced 2 collisions.
dot3Collided3Times	Number of frames that experienced 3 collisions.
dot3Collided4Times	Number of frames that experienced 4 collisions.
dot3Collided5Times	Number of frames that experienced 5 collisions.
dot3Collided6Times	Number of frames that experienced 6 collisions.
dot3Collided7Times	Number of frames that experienced 7 collisions.
dot3Collided8Times	Number of frames that experienced 8 collisions.
dot3Collided9Times	Number of frames that experienced 9 collisions.
dot3Collided10Times	Number of frames that experienced 10 collisions.
dot3Collided11Times	Number of frames that experienced 11 collisions.
dot3Collided12Times	Number of frames that experienced 12 collisions.
dot3Collided13Times	Number of frames that experienced 13 collisions.
dot3Collided14Times	Number of frames that experienced 14 collisions.
dot3Collided15Times	Number of frames that experienced 15 collisions.

Class of Service Statistics

The class of service statistics shown in Table 52 are generated by the Send Data Initiator state machine.

Table 52: Send Data Initiator Class of Service Statistics

Value Name	Description
COSIfHCOutPkts[1-16]	Number of frames sent on each of the 16 send rings.

The class of service statistics shown in Table 53 are generated by the Receive List Placement state machine.

Table 53: Receive List Placement Class of Service Statistics

Value Name	Description
COSFramesDroppedDueToFilters	
COSIfHCInPkts[1-16]	Number of frames received and placed on each of the 16 receive return rings.

STATISTICS MEMORY BLOCK

The statistics memory region between 0x300 and 0xaff in the NIC local memory (see Table 54) can be accessed from several modes: Flat, Standard, or Indirect. For a discussion of these modes, see Section 9: "PCI" on page 178. This statistics memory block does not exist in the BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 controllers.

Table 54: Statistics Block Memory

Offset	Description	Access
0x300-0x3ff	Reserved	-
0x400	ifHCInOctets	R/W
0x408	Reserved	R/W
0x410	etherStatsFragments	R/W
0x418	ifHCInUcastPkts	R/W
0x420	ifHCInMulticastPkts	R/W
0x428	ifHCInBroadcastPkts	R/W
0x430	dot3StatsFCSErrors	R/W
0x438	dot3StatsAlignmentErrors	R/W
0x440	xonPauseFramesReceived	R/W
0x448	xoffPauseFramesReceived	R/W
0x450	macControlFramesReceived	R/W
0x458	xoffStateEntered	R/W
0x460	dot3StatsFrameTooLongs	R/W
0x468	etherStatsJabbers	R/W
0x470	etherStatsUndersizePkts	R/W
0x478	inRangeLengthError	R/W
0x480	outRangeLengthError	R/W
0x488	etherStatsPkts64Octets	R/W
0x490	etherStatsPkts65Octetsto127Octets	R/W
0x498	etherStatsPkts128Octetsto255Octets	R/W
0x4a0	etherStatsPkts256Octetsto511Octets	R/W
0x4a8	etherStatsPkts512Octetsto1023Octets	R/W
0x4b0	etherStatsPkts1024Octetsto1522Octets	R/W
0x4b8	etherStatsPkts1523Octetsto2047Octets	R/W
0x4c0	etherStatsPkts2048Octetsto4095Octets	R/W
0x4c8	etherStatsPkts4096Octetsto8191Octets	R/W
0x4d0	etherStatsPkts8192Octetsto9022Octets	R/W
0x4d8-0x5ff	Reserved	-
0x600	ifHCOutOctets	R/W
0x608	Reserved	-
0x610	etherStatsCollisions	R/W
0x618	outXonSent	R/W
0x620	outXoffSent	R/W
0x628	flowControlDone	R/W
0x630	dot3StatsInternalMacTransmitErrors	R/W
0x638	dot3StatsSingleCollisionFrames	R/W
0x640	dot3StatsMultipleCollisionFrames	R/W

Table 54: Statistics Block Memory (Cont.)

Offset	Description	Access
0x648	dot3StatsDeferredTransmissions	R/W
0x650	Reserved	-
0x658	dot3StatsExcessiveCollisions	R/W
0x660	dot3StatsLateCollisions	R/W
0x668	dot3Collided2Times	R/W
0x670	dot3Collided3Times	R/W
0x678	dot3Collided4Times	R/W
0x680	dot3Collided5Times	R/W
0x688	dot3Collided6Times	R/W
0x690	dot3Collided7Times	R/W
0x698	dot3Collided8Times	R/W
0x6a0	dot3Collided9Times	R/W
0x6a8	dot3Collided10Times	R/W
0x6b0	dot3Collided11Times	R/W
0x6b8	dot3Collided12Times	R/W
0x6c0	dot3Collided13Times	R/W
0x6c8	dot3Collided14Times	R/W
0x6d0	dot3Collided15Times	R/W
0x6d8	ifHCOutUcastPkts	R/W
0x6e0	ifHCOutMulticastPkts	R/W
0x6e8	ifHCOutBroadcastPkts	R/W
0x6f0	dot3StatsCarrierSenseErrors	R/W
0x6f8	ifOutDiscards	R/W
0x700	ifOutErrors	R/W
0x708-0x7ff	Reserved	-
0x800-0x87f	COSIfHCInPkts[1-16]	R/W
0x880	COSFramesDroppedDueToFilters	R/W
0x888	nicDmaWriteQueueFull	R/W
0x890	nicDmaWriteHighPriQueueFull	R/W
0x898	nicNoMoreRXBDs	R/W
0x8a0	ifInDiscards	R/W
0x8a8	ifInErrors	R/W
0x8b0	nicRecvThresholdHit	R/W
0x8b8-0x8ff	Reserved	-
0x900-0x97f	COSIfHCOutPkts[1-16]	R/W
0x980	nicDmaReadQueueFull	R/W
0x988	nicDmaReadHighPriQueueFull	R/W
0x990	nicSendDataCompQueueFull	R/W
0x998	nicRingSetSendProdIndex	R/W
0x9a0	nicRingStatusUpdate	R/W
0x9a8	nicInterrupts	R/W
0x9b0	nicAvoidedInterrupts	R/W
0x9b8	nicSendThresholdHit	R/W
0x9c0-0xaff	Reserved	-

Section 6: Receive Data Flow

INTRODUCTION

The RX MAC pulls BDs from RX producer rings. The RX BD specifies the location(s) in host memory where packet data may be moved. Figure 47 on page 117 shows the receive buffer descriptor cycle.

All ingress Ethernet frames are classified by the RX rules engine. A class ID is associated to each frame based on QOS rules setup in the RX MAC (see "Receive Rules Setup and Frame Classification" on page 126). The Receive List Placement and Receive List Initiator portions of the MAC architecture move BDs to the RX return rings; the class ID associated to the packet is examined to route the BD to a specific RX return ring.

The BCM5700, BCM5701, BCM5702, BCM5703C, BCM5703S, BCM5704C, and BCM5704S devices support sixteen RX Return Rings. The sixteen RX Return rings allow host software to categorize traffic groups based on the QOS rules. The BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714C, BCM5714S, BCM5715C, and BCM5715C devices support only one Rx Return Ring.

Once the packet is queued to the RX return ring, the device driver will wait for indication of the same through the status block update and interrupt from the host coalescing engine. The host coalescing engine will update the status block and generate a line interrupt or MSI (see "Host Coalescing" on page 287 for further details regarding interrupts) when a specified host coalescence criteria is met. Once the interrupt is generated, the host device driver will service the interrupt. The ISR will determine if new BDs have been completed on the RX Return Rings. Next, the device driver will indicate to the network protocol that the completed RX packets are available. The network protocol will consume the packets and return physical buffers to the network driver at a later point.

The BDs may then be reused for new RX frames. The device driver must return the BD to an RX producer ring. For this purpose, the driver should fill out either the opaque field or index field of the Rx BD when inserting/initializing the BD in an RX Producer ring. When the BD is returned by the device through Return Ring, the opaque or index data field of the BD will be used by the driver to identify the BD in Producer Ring that corresponds to the Returned BD in Return Ring. The device driver will then reinitialize the identified BD in Producer Ring with a new allocated buffer and replenish the Receive Producer Ring with this BD.

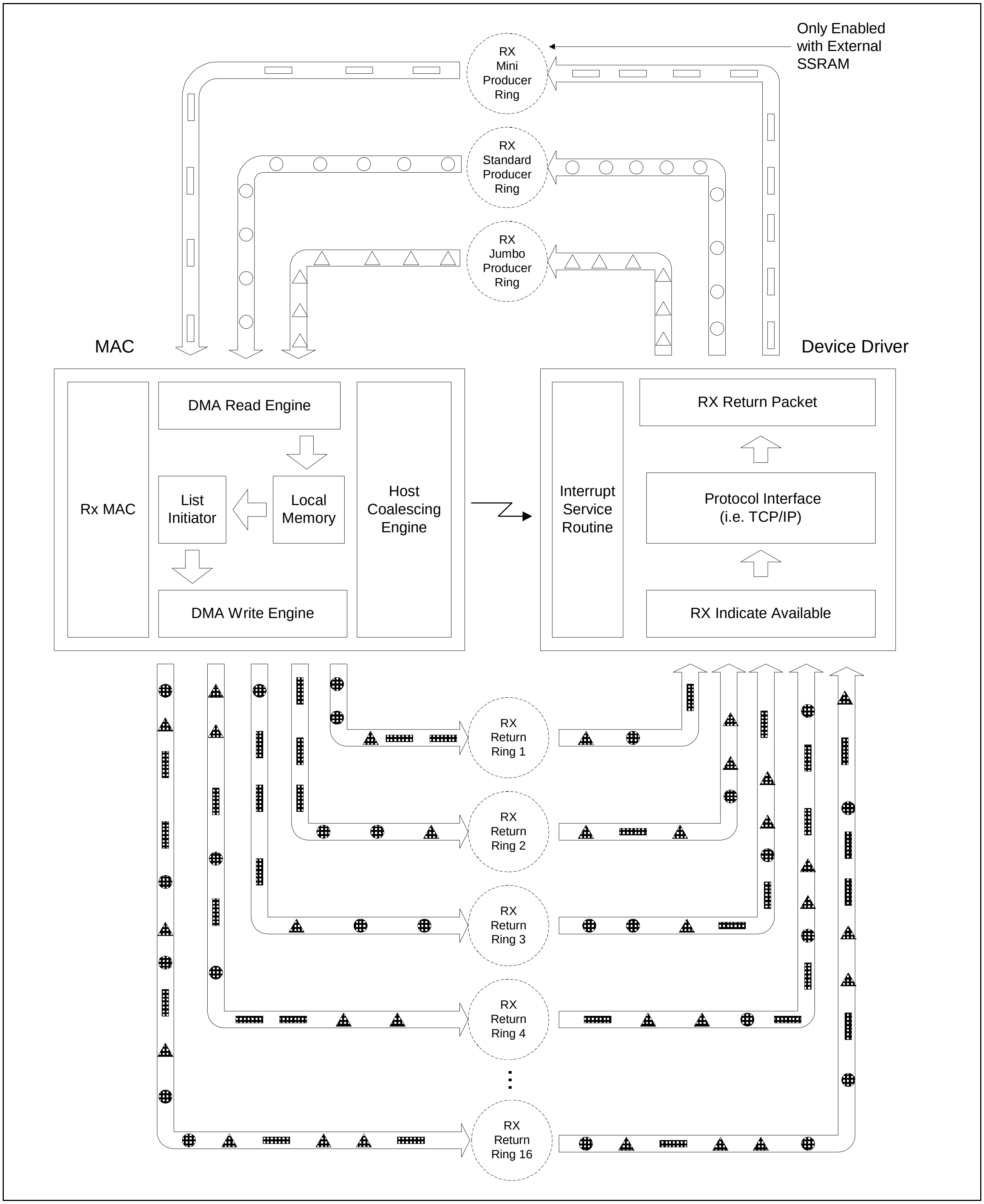


Figure 47: Receive Buffer Descriptor Cycle

RECEIVE PRODUCER RING

A Receive Producer Ring is an array containing a series of Receive Buffer Descriptors (BD). There are up to three Receive Producer Rings (PR) supported by at least one member of the BCM57XX family (see Table 55); the Mini PR, the Jumbo PR, and the Standard PR. The three Receive Producer Rings will help efficient host memory utilization by providing the device with buffers of three different sizes. The Receive Producer Rings are host-based and 25% of the available buffer descriptors are cached in BCM57XX internal memory.

Table 55: Receiver Producer Rings Supported by BCM57XX Family

	Standard Ring	***Jumbo Ring***	***Mini Ring***
BCM5700 with only internal SSRAM	Supported	Supported	Not supported
BCM5700 with external SSRAM	Supported	Supported	Supported
BCM5701, BCM5702, BCM5703C, BCM5703S, BCM5704C, and BCM5704S	Supported	Supported	Not supported
BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714C, BCM5714S, BCM5715C, and BCM5715S	Supported	Not supported	Not supported

The use of the Mini and Jumbo producer rings is optional; the Standard is required.

- Only BCM5700 with external SSRAM supports the Mini Ring. This is because a copy of each ring is cached in BCM5700 internal memory and the BCM5700 MAC expects the Mini Producer Ring addresses at a higher memory than is supported with internal memory.

A receive producer ring contains a series of buffer descriptors which in turn contain information of host memory locations to where packets are placed by the BCM57XX family at reception. Each receive producer ring has a limit to the amount of buffer descriptors it can contain as follows:

- Standard producer ring is limited to a maximum capacity of 512 BDs (see Table 32 on page 98).
- Mini producer ring is limited to a maximum of 1024 BDs (see Table 31 on page 97).
- Jumbo producer ring is limited to a maximum of 256 BDs (see Table 33 on page 98).

The Receive Producer Rings cannot be concatenated and the number of buffer descriptors in each ring is limited to the maximum buffer descriptor capacity of that ring. The BCM57XX family keeps copies of 25% of the buffer descriptors of each ring in its internal memory. It is not possible to set up three rings of one type or use any different combination of Return Producer Rings other than what has been specified.

Set Up of Producer Rings Using RCBs

A Ring Control Block (RCB) is used by the host software to set up the shared rings in host memory. In the context of producer rings, a RCB is a register that is used to define a single Receive Producer Ring. Each producer ring has a single RCB that corresponds to it, which describes the ring and is used during setup and initialization. The host software must initialize these registers for each producer ring.

The Jumbo, Standard, and Mini producer rings have a corresponding RCB register called:

- Jumbo Receive BD ring RCB—register offset 0x2440-0x244f
- Standard Receive BD ring RCB—register offset 0x2450-0x245f
- Mini Receive BD ring RCB—register offset 0x2460-0x246f

Each Receive Producer Ring RCB register is 16 bytes long and is found in the Receive BD Initiator Control registers—memory offset 0x2400 (see "Ring Control Blocks" on page 91).

Other Considerations Relating to Producer Ring Setup

Other registers that affect the producer rings need to be initialized by the host software. These registers include the Receive BD Ring Replenish Threshold register, the Receive MTU register, and the Accept Oversized bit (bit 5) in the Receive MAC Mode register.

- Receive BD Producer Ring Replenish Threshold registers:
  - "Mini Receive BD Producer Ring Replenish Threshold Register (Offset 0x2C14)" on page 443.
  - "Standard Receive BD Producer Ring Replenish Threshold Register (Offset 0x2C18)" on page 443.
  - "Jumbo Receive BD Producer Ring Replenish Threshold Register (Offset 0x2C1C)" on page 443.

  These registers are used for setting the number of BDs the BCM57XX family can use up before requesting more BDs be DMAed from a producer ring. In other words, the device will not initiate a DMA for fetching the Rx BDs until the number of BDs made available to the device by the host is at least the value programmed in this register.
- Receive MTU register ("Receive MTU Size Register (Offset 0x43C)" on page 387). This 32-bit register is set to a value that is the maximum size of a packet that the BCM57XX family will receive. Any packets above this size is labeled as an oversized packet. The value for this register is typically set to 1518, which is the Standard Producer Ring RCB typical setting. If Jumbo frames are supported, the MTU would be set to the maximum Jumbo frame size.
- Receive MAC Mode register ("Receive MAC Mode Register (Offset 0x468)" on page 391). If the Accept Oversized bit (bit 5) of this register is set, the BCM57XX family accepts packets larger than specified in the MTU (up to 64k bytes).

RCB Setup Pseudo Code

An example of setting up a standard receive producer ring using the RCB:

```
Content of Pointer_to_Std_RX_RCB + 0x00 = Host address of standard receive producer ring
high 32.
Content of Pointer_to_Std_RX_RCB + 0x04 = Host address of standard receive producer ring low
32.
Content of Pointer_to_Std_RX_RCB + 0x0a = No flags.
Content of Pointer_to_Std_RX_RCB + 0x08 = Max packet size of 1518.
Content of Pointer_to_Std_RX_RCB + 0x0c = Internal Memory address for device copy of ring.
```

Figure 48 on page 120 shows the standard ring RCB for the setup of a host-based standard producer ring.

Receive Buffer Descriptors (BDs) begin on the Receive Producer Rings. The host device driver will populate each ring with a specified number of BDs supported by the ring type: jumbo, standard, or mini (see "Receive Producer Ring" on page 118). The Mini Producer Ring is only available when external SSRAM is attached to the MAC. When the host device driver enables the Mini Producer Ring, the memory region for packet buffers must be moved to another location in external memory. The Mini Producer Ring displaces internal packet buffers (see "Initialization" on page 146). When a packet is received, the RX MAC moves the packet data into internal memory. The size of the packet will determine which RX Producer Ring is the BD source bucket. The Ring Control Blocks (RCB) for the Mini and Standard RCB contain a bit field called Max_Len (see "Ring Control Blocks" on page 91). Host software must program a threshold value into the Max_Len field. The threshold value specifies the appropriate ring that the RX MAC must use to source a BD, based on ingress packet size. The Jumbo RCB ignores the Max_Len field. The Receive MTU Size register (see "Receive MTU Size Register (Offset 0x43C)" on page 387) specifies the largest packet accepted by the RX MAC; packets larger than the Receive MTU are marked oversized and are discarded. Figure 49 on page 121 shows how RX Producer rings can be programmed to associate to packet size.

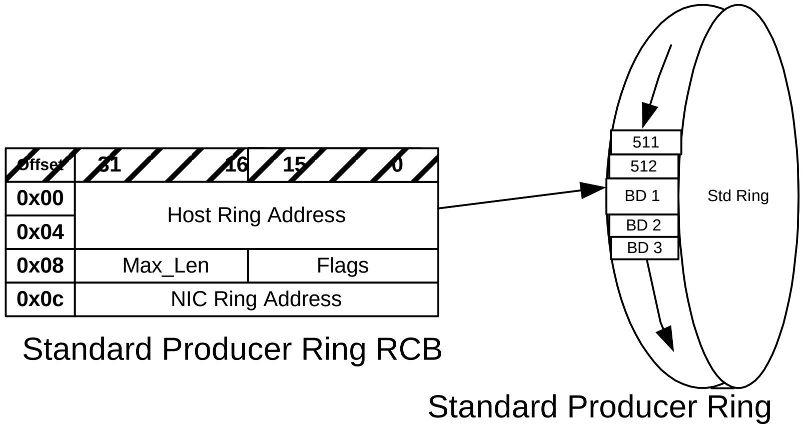


Figure 48: Standard Ring RCB for Setup of a Host-based Standard Producer Ring

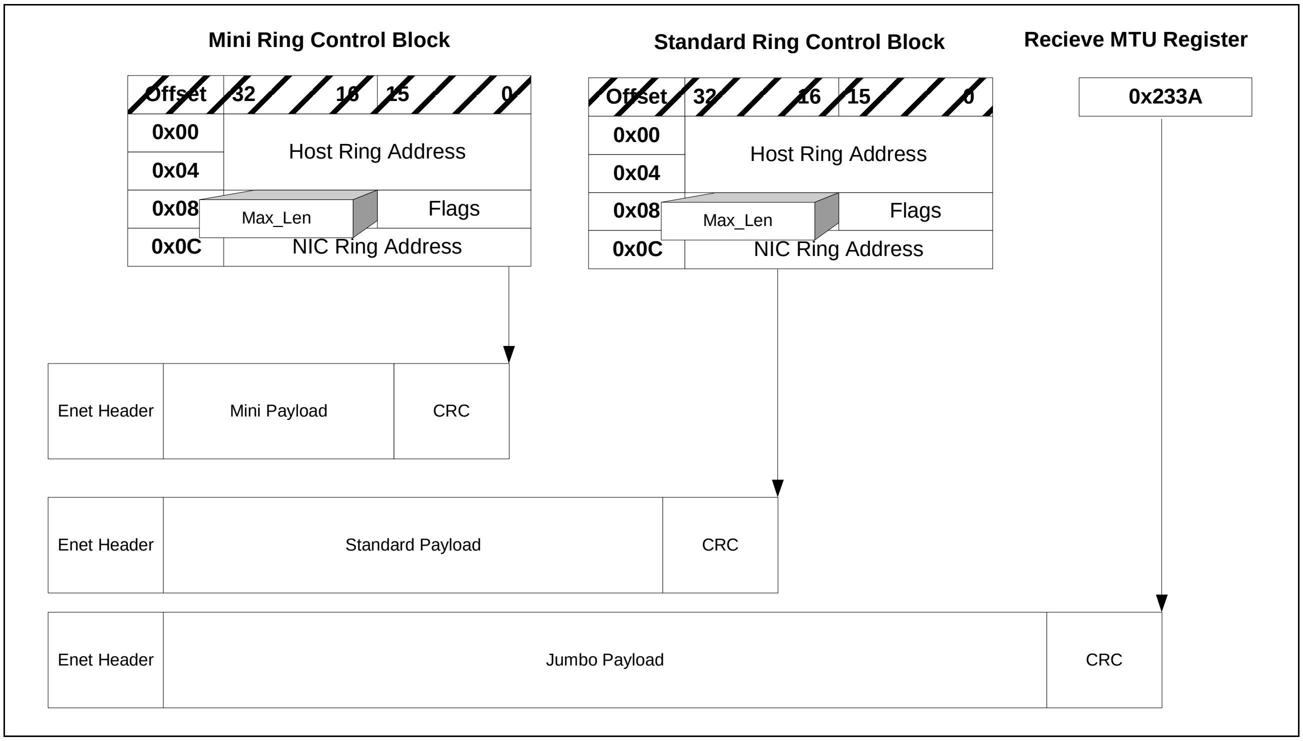


Figure 49: Frame Sizes

RECEIVE BUFFER DESCRIPTORS

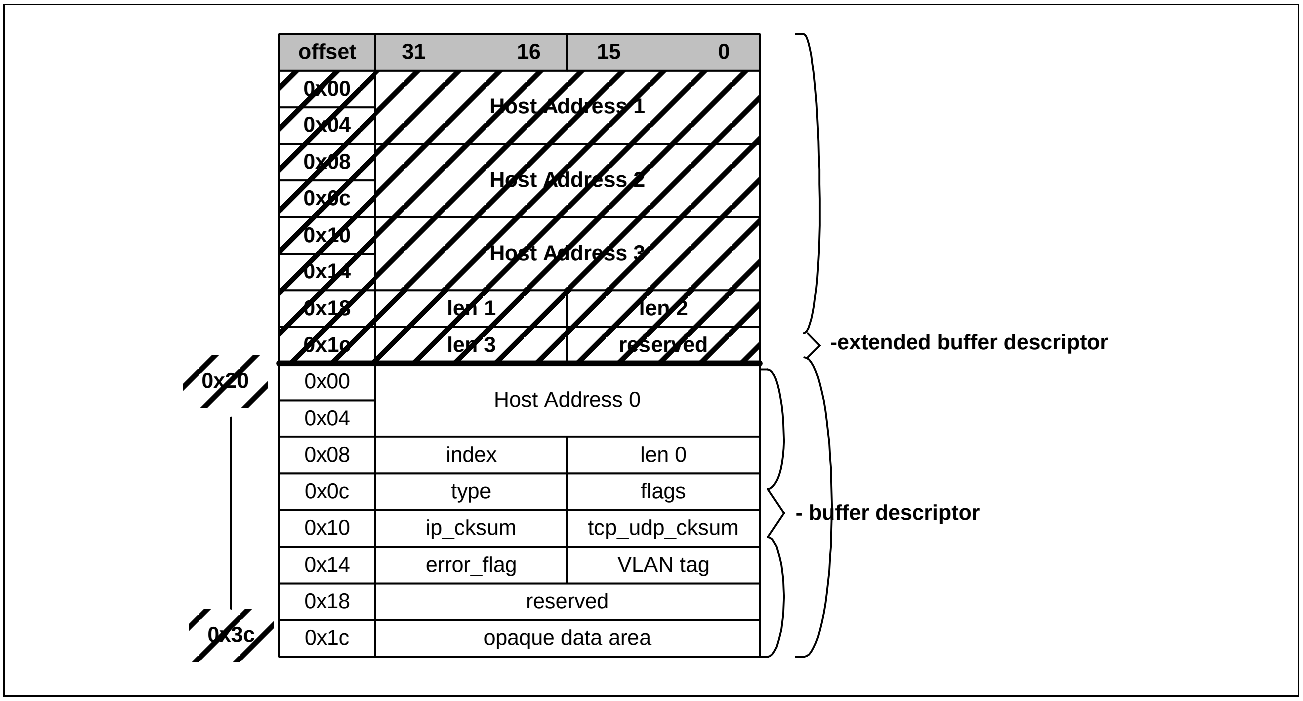
The Receive Buffer Descriptor is a data structure in host memory. It is the basic element that makes up each receive producer and receive return ring. The format of receive buffer descriptors can be seen in Figure 50. There is also an extended receive buffer descriptor format that is optionally used for supporting the Jumbo Frames. If the extended receive buffer descriptor is used, the device always returns only the non-extended portion of a receive buffer descriptor to the host in the receive return ring.

A receive buffer descriptor has a 64-bit memory address and may be in any memory alignment and may point to any byte boundary. For performance and CPU efficiency reasons, it is recommended that memory be cache-aligned. The BCM57XX family supports cache line sizes of 8, 16, 32, 64, 128, 256, and 512 bytes. The cache line size value is important for the controller to determine when to use the PCI memory write and invalidate command. There are no requirements for memory alignment or cache line integrity for the BCM570X.

Unlike send buffer descriptors, the receive buffer descriptors cannot be chained to support multiple fragments. Multiple fragment support can only be accomplished by using extended buffer descriptors.

Extended Receive Buffer Descriptor

The format of the extended receive buffer descriptor is shown in Figure 50. The only differences between the extended receive buffer descriptor and a normal receive buffer descriptor are the additional *Host address n* fields that contain the address of the nth piece of a fragment in host memory and the corresponding *len n* field that holds the length of the data pointed to by the *host address n.* Please see "Receiving Jumbo Frames" on page 133 for the description of how extended BDs are used for receiving Jumbo Frames.



offset	31 16	15 0
0x00	Host Address 1	
0x04		
0x08	Host Address 2	
0x0c		
0x10	Host Address 3	
0x14		
0x18	len 1	len 2
0x1c	len 3	reserved
0x00	Host Address 0	
0x04		
0x08	index	len 0
0x0c	type	flags
0x10	ip_cksum	tcp_udp_cksum
0x14	error_flag	VLAN tag
0x18	reserved	
0x1c	opaque data area	

Figure 50: Receive Buffer Descriptor and Extended Buffer Descriptor

In using the extended buffer descriptor, the host software can use multiple receive fragments to support jumbo-sized packets, up to 9 KB. The host software is responsible for keeping track of the fragment addresses and lengths because the BCM57XX family will return only the non-extended portion of the buffer descriptor to the return rings and the len 0 field will contain the total size of the packet.

Management of Rx Producer Rings with Mailbox Registers and Status Block

Status Block

The host software manages the producer rings through the Mailbox registers and by using the Status Block. It does this by writing to the Mail Box registers when a BD is available to DMA to the BCM57XX family and reading the Status Block to see how many BDs have been consumed by the BCM57XX family (see Figure 51 on page 124). The Status Block can be seen in Table 39 on page 105.

The Status Block is controlled and updated by the BCM57XX family. The status block in host memory is constantly updated through a DMA copy by the BCM57XX family from an internal Status Block. The updates occur at specific intervals and host coalescence conditions that are specified by host software during initialization of the BCM57XX family. The registers for setting the intervals and conditions are in the Host Coalescing Control registers (see "Host Coalescing" on page 62) starting at memory offset 0x3c00. The BCM57XX family will DMA an updated Status Block to the 32-bit address that is set by the host software in the Host Coalescing Control registers, register 0x3c38.

Among other status, the Status Block displays the last 16-bit value, BD index that was DMAed to the BCM57XX family from each producer ring. The BCM57XX family updates these indices as the recipient or consumer of the BD from the producer rings. The updated indices for the Standard, Jumbo, and Mini producer rings are the recv std cons, recv jumbo cons, and recv mini cons indices.

Mailbox

The host software is responsible for writing to the Mailbox registers (see Figure 51) when a BD is available from the producer rings for use by the BCM57XX family. Host software should use the high-priority mailbox region from 0x200 through 0x3FF for host standard and flat modes and the low-priority mailbox region from 0x5800 through 0x59FF for indirect register access mode.

The Mailbox registers (starting at memory offset 0x200 for host standard and flat modes and offset 0x5800 for indirect mode) contain the following indices:

- Receive BD Standard Producer Ring Producer Index
  - Host standard and flat modes: memory offset 0x268–0x26F
  - Indirect mode: memory offset 0x5868–0x586F
- Receive BD Jumbo Producer Ring Producer Index
  - Host standard and flat modes: memory offset 0x270–0x277
  - Indirect mode: memory offset 0x5870–0x5877
- Receive BD Mini Producer Ring Index
  - Host standard and flat modes: memory offset 0x278–0x27F
  - Indirect mode: memory offset 0x5878–0x587F

Offset (High-Priority Mailboxes for Host Standard and Flat Modes)	Offset (Low-Priority Mailboxes for Indirect Mode)	Register	Access
0x200 - 0x207	0x5800 - 0x5807	Interrupt Mailbox 0	RW
0x208 - 0x20F	0x5808 - 0x580F	Interrupt Mailbox 1	RW
▨	▨	▨	▨
▨	▨	▨	▨
0x268 - 0x26F	0x5868 - 0x586F	Receive BD Standard Producer Ring Producer Index	RW
0x270 - 0x277	0x5870 - 0x5877	Receive BD Jumbo Producer Ring Producer Index	RW
0x278 - 0x27F	0x5878 - 0x587F	Receive BD Mini Producer Ring Producer Index	RW
0x280 - 0x287	0x5880 - 0x5887	Receive BD Return Ring 1 Consumer Index	RW
0x288 - 0x28F	0x5888 - 0x588F	Receive BD Return Ring 2 Consumer Index	RW
0x290 - 0x297	0x5890 - 0x5897	Receive BD Return Ring Consumer Index	RW
▨	▨	▨	▨

Figure 51: Mailbox Registers

Each register contains the index value of the next buffer descriptor from the corresponding producer ring that is available for DMA to the BCM57XX family from the host. When the host software updates one of these indices, the BCM57XX family is automatically signaled that a new BD is waiting for DMA. At initialization time, these values must be initialized to zero. These indices are 64-bit wide; however, the highest index value is only 1024 for the Mini Producer Ring.

Receive Return Rings

Receive Return Rings (RR) are host-based memory blocks that are used by host software to keep track of the where the BCM57XX family is putting the received packets related receive buffer descriptors. Unlike the producer rings, the return rings reside only in host memory. The device supports 16 Receive Return Rings regardless of whether external memory is present.

The BCM57XX family uses the BDs in the NIC's memory that are previously copied from the producer rings to use when packets are received from the LAN. It places the BDs that correspond to received packets in the return rings.

Return rings are the exact opposite of producer rings, except that they are not categorized by the maximum length receive packets supported. They are actually categorized by priority or class of received packet. The highest priority return ring is ring 1, and the lowest priority is the last ring (Return Ring 2–Return Ring 16 depending on how many rings are set up by the host software). See Receive Rules setup and Frame Classification below. Return rings come in only one of two maximum buffer descriptor capacity sizes:

- 1024 is valid only if RCB_FLAG_RING_DISABLED flag is set in Mini Producer Ring RCB.
- 2048 buffer descriptors if the Mini producer ring is supported.

The BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714C, BCM5714S, BCM5715C, and BCM5715S devices support only one Receive Return Ring and this ring size is configurable to a value of up to 512.

RCBs are used to set up return rings in much the same way they are used to set up the producer rings. These RCBs for the return rings are set in the Miscellaneous memory region (SSRAM) at offset 0x200 (this region should not be confused with the register space in the chip). The RCB max_len field is used to indicate the number of buffer descriptor entries in a return ring. If an invalid value is set, the BCM57XX family will indicate an attention error in the Flow Attention register. Figure 45 on page 96 shows receive return rings.

Management of Return Rings with Mailbox Registers and Status Block

The return rings are managed by the host using the Mailbox registers and Status Block.

When a packet is received from the LAN, the BCM57XX DMAs the packet to a location in the host, and then DMAs the related BD to a return ring. As the producer of this packet to the host, the BCM57XX family updates the Status Block producer indices for the related return ring (i.e., return ring 1 to return ring 16 that was DMAed the BD received packet). These return ring indices, RX Return Producer 1 to RX Return Producer 16 shown in Table 39 on page 105, can then be read by the host software to determine the last BD index value of a particular ring that has information of the last received packet.

As the consumer of the received packet, the host software must update the return ring consumer indices in Mailbox registers Receive BD Return Ring 1 Consumer Index (memory offset 0x280–0x287 for host standard and flat modes and 0x5880–0x5887 for indirect mode) through Receive BD Return Ring 16 Consumer Index (memory offset 0x300–0x307 for host standard and flat modes and 0x5900–0x5907 for indirect mode).

Host Buffer Allocation

The allocation of memory in the host is dependent on the operating system in which the controller is being used. Two things to remember are:

- The use of non-cached and physically contiguous memory is best for adapter performance.
- Physical memory mapping is needed for the controllers internal copies of logical host memory.

Receive Rules Setup and Frame Classification

The BCM57XX family has a feature that allows for the classification of receive packets based on a set of rules. The rules are determined by the host software and then input into the BCM57XX family.

A packet can be accepted or rejected based on the rules initialized into two rules register areas. The packets can also be classified into groups of packets of higher to lower priority using the rules registers. This occurs when the packet is directed to a specific return ring. Return rings 1-16 have an inherent priority associated with them. The priority is from lowest ring number to highest ring number; return ring 1 being the highest priority ring and return ring 16 being the lowest. The implementation of priority class is based on how many rings the host software has initialized and made available to the BCM57XX family. As packets arrive, the BCM57XX family may classify each packet based on the rules. When the host services the receive packet, it can service the lower numbered rings first.

A rule can be changed by first disabling it by setting 0 into Enable bit (bit 31) in Receive BD Rules Control register (see Table 57). Wait about 20 receive clocks (rx_clock) and then re-enable it when it is programmed with a new rule. Otherwise, changing the rules dynamically during runtime may cause the rule checker to output erroneous results because the rule checker is a pipelined design and uses the various fields of the rules at different clock cycles.

Receive Rules Configuration Register

The Receive Rules Configuration register (memory offset 0x500–0x503, see Table 56) uses bits 3-7 to specify the ring where a received packet should be placed into if no rules are met, or if the rules have not been set up. A value of 0 means the received packet will be discarded. A value of 1-16 specifies a corresponding ring. This ring should be initialized to at least a value of 1 if the rules are not being used to ensure that all received packets will be DMAed to return ring 1.

Table 56: Receive Rules Configuration Register

Bits	*Field*	*Access*
31-8	Reserved	R/O
7-3	Specifies the default class (ring) if no rules are matched	R/W
2-0	Reserved	R/O

Receive List Placement Rules Array

The Receive List Placement Rules Array (memory offset 0x480–0x4ff) is made up of 16 combined element registers. The combined element is actually two 32-bit registers called the Receive BD Rules Control register (see Table 57) and the Receive BD Rules Value/Mask register (see Table 58). The element can be looked at as a single 64-bit entity with a Control part and Value/Mask part since they use a single element. Bit 26 of the control part determines how the value/mask part is used. The Receive BD Rules Value/Mask register, or Value/Mask can be used as either a 32-bit left-justified value or a 16-bit mask followed by a 16-bit value.

Note: Receive rules cannot be used to match VLAN headers because the VLAN tag is stripped from the Ethernet frame before the rule checker runs.

Table 57: Receive BD Rules Control Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E	&	P1	P2	P3	M	D	Map	Reserved						Op	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Header			Class					Offset							

Bit	*Name*	*R/W*	*Description*	*Default*
31	E	R/W	Enable. Enabled if set to 1	-
30	&	R/W	And With Next. This rule and next must both be true to match. The class fields must be the same. A disabled next rule is considered true. Processor activation bits are specified in the first rule in a series.	-
29	P1	R/W	If the rule matches, the processor is activated in the queue descriptor for the Receive List Placement state machine.	-
28	P2	R/W	If the rule matches, the processor is activated in the queue descriptor for the Receive Data and Receive BD Initiator state machine.	-
27	P3	R/W	If the rule matches, the processor is activated in the queue descriptor for the Receive Data Completion state machine.	-
26	M	R/W	Mask If set, specifies that the value/mask field is split into a 16-bit value and 16-bit mask instead of a 32-bit value.	-
25	D	R/W	Discard Frame if it matches the rule.	-
24	Map	R/W	Map Use the masked value and map it to the class.	-
23:18	Reserved	R/W	Reserved bits must be written to zero.	0
17:16	Op	R/W	Comparison Operator specifies how to determine the match: • 00 = Equal • 01 = Not Equal • 10 = Greater than • 11 = Less Than	-

Bit	Name	R/W	Description	Default
15:13	Header	R/W	Header Type specifies which header the offset is for: • 000: Start of Frame (always valid) • 001: Start of IP Header (if present) • 010: Start of TCP Header (if present) • 011: Start of UDP Header (if present) • 100: Start of Data (always valid, context sensitive) • 101–111: Reserved	-
12:8	Class	R/W	The class this frame is placed into if the rule matches. 0-16, where 0 means discard. The number of valid classes is the Number of Active Queues divided by the Number of Interrupt Distribution Groups. Ring 1 has the highest priority and ring 16 has the lowest priority.	-
7:0	Offset	R/W	Number of bytes offset specified by the header type.	-

Table 58: Receive BD Rules Value/Mask Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Mask															

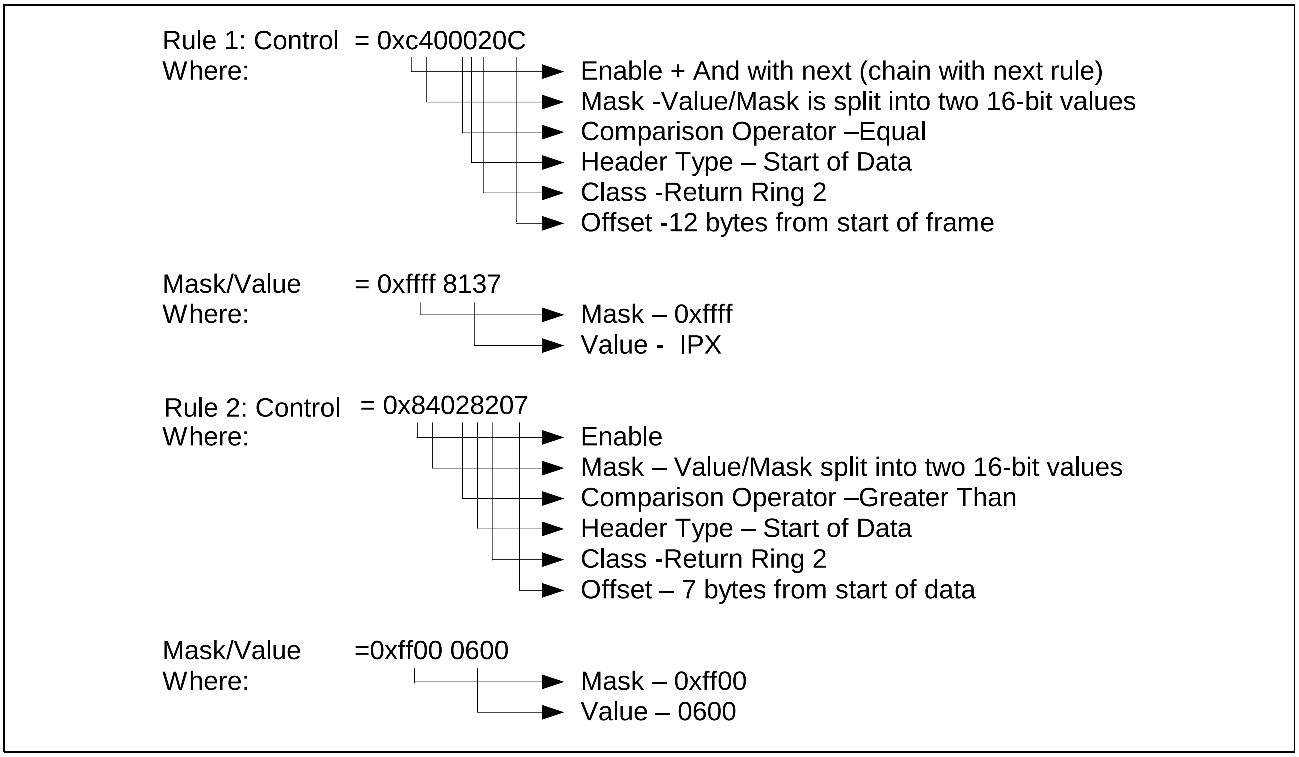
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Value															

Bit	Name	R/W	Description	Default
31:16	Mask	-	-	-
15:0	Value	-	-	-

Class of Service Example

If either Start of IP Header, Start of TCP Header, or Start of UDP Header is specified, and the frame has no IP, TCP, or UDP header, respectively, there is no frame match. The full set of rules provides a fairly rich selection and filtering criteria.

> ***Example:*** If you wanted to set a Class of Service (CoS) of 2 based on the eighth byte in the data portion of an encapsulated IPX frame using Ethernet Type 2 having a value greater than 6, you could set up the rules shown in Figure 52.



Rule 1: Control = 0xc400020C
Where:
- Enable + And with next (chain with next rule)
- Mask -Value/Mask is split into two 16-bit values
- Comparison Operator –Equal
- Header Type – Start of Data
- Class -Return Ring 2
- Offset -12 bytes from start of frame

Mask/Value = 0xffff 8137
Where:
- Mask – 0xffff
- Value - IPX

Rule 2: Control = 0x84028207
Where:
- Enable
- Mask – Value/Mask split into two 16-bit values
- Comparison Operator –Greater Than
- Header Type – Start of Data
- Class -Return Ring 2
- Offset – 7 bytes from start of data

Mask/Value =0xff00 0600
Where:
- Mask – 0xff00
- Value – 0600

Figure 52: Class of Service Example

Checksum Calculation

Whether the host software NOS supports checksum offload or not, the BCM57XX family automatically calculates the IP, TCP, and UDP of received packets as described in RFC 791, RFC 793, and RFC 768 respectively.

Which protocol checksum value is produced can be determined by reading the status flag field in the Receive Return Ring. The valid flag values in the status flag field are IP_CHECKSUM and TCP_UDP_CHECKSUM. When a valid checksum is produced, the values of the checksums are found in the corresponding receive buffer descriptor register. These values should be 0xFFFF for a valid checksum or any other value if the checksum was incorrectly calculated. Assert the Receive No Pseudo-header Checksum bit of the Mode Control register (see "Mode Control Register (Offset 0x6800)" on page 502) to disable TCP/UDP pseudo-header checksums.

VLAN Tag Strip

Receiving VLAN-tagged (IEEE 802.1q compliant) packets are automatically supported by the BCM57XX family. There is no register or setting needed to receive packets that are VLAN-tagged. The VLAN tag is automatically stripped from the 802.1q compliant packet at reception and then placed in a receive buffer descriptor's two byte VLAN tag field. The flag field has the BD_FLAGS_VLAN_TAG bit set when a valid VLAN packet is received. Once the packet has been serviced by the host software, these fields should be zeroed out.

In the Receive MAC Mode register—memory offset 0x468–0x46b, the Keep VLAN Tag Diag Mode bit (bit 10) can be set to force the BCM57XX family to not strip the VLAN tag from the packet. This is only for diagnostic purposes.

The following table shows the frame format with 802.1Q VLAN tag inserted.

Table 59: Frame Format with 802.1Q VLAN Tag Inserted

Offset	*Description*
0-5	MAC destination address
6-11	MAC source address
12-13	Tag Protocol ID (TPID)—0x8100
14-15	Tag Control Information (TCI): • Bit 15-13: 802.1P priority • Bit 12: CFI bit • Bit 11-0: VLAN ID
16-17	The original Ethertype
18-1517	Payload

RX DATA FLOW DIAGRAM

The receive data flow can be summarized in Figure 53. The Receive Producer Rings, Receive Buffer Descriptors, Receive Return Rings, Mailbox registers, and Status Block registers are the main areas of the receive data flow.

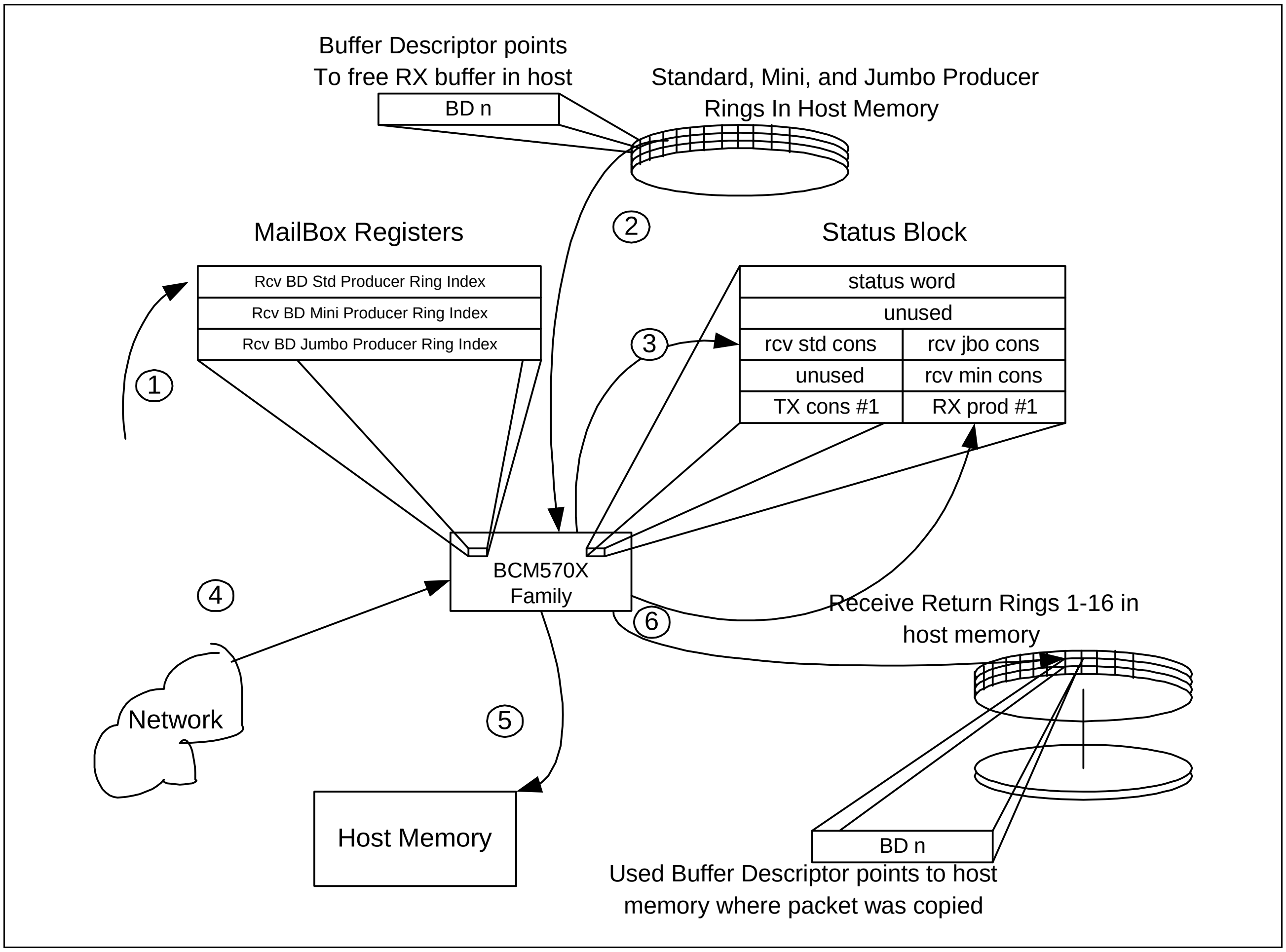

Buffer Descriptor points
To free RX buffer in host
BD n
Standard, Mini, and Jumbo Producer
Rings In Host Memory
MailBox Registers
2
Status Block
Rcv BD Std Producer Ring Index
Rcv BD Mini Producer Ring Index
Rcv BD Jumbo Producer Ring Index
status word
unused
3
rcv std cons
rcv jbo cons
1
unused
rcv min cons
TX cons #1
RX prod #1
BCM570X
Family
4
Receive Return Rings 1-16 in
host memory
6
Network
5
Host Memory
BD n
Used Buffer Descriptor points to host
memory where packet was copied

Figure 53: Overview Diagram of RX Flow

The following steps describe the RX flow sequence.

1. The host software updates a Receive Producer Ring Index in the Mailbox registers.
2. A receive BD or series of BDs with the corresponding index is DMAed to BCM57XX family from the host-based Receive Producer Ring.
3. The BCM57XX family updates the Receive Consumer Index in the Host Block register and stores copy of the BD.
4. A valid Ethernet packet is received from the network into the B5700.
5. The Ethernet packet is DMAed to host memory using a BD previously DMAed from a Receive Producer Ring.
6. The BD used for the received packet is DMAed from the BCM57XX family to one of the Receive Return Rings, and the Receive Return Ring Producer index register in the Host Status Block is updated by the BCM57XX family.

The host software must create an array of BD structures in host memory, referred to as a Receive Producer Ring. Each receive buffer descriptor within a producer ring describes, among other things, the location of a host memory buffer that is used to store the packets received from the Network. When the host software (as the producer) updates the Mailbox register's producer ring index that corresponds to the Receive Producer Ring, the BCM57XX family automatically DMAs the BD to itself (offset 0x6000 for Standard BD and offset 0x7000 for Jumbo BD) from the host. When the DMA is completed, the BCM57XX family (as the consumer) updates the Status Block's Receive Consumer Ring Index to signal it successfully consumed the BD. The BCM57XX family keeps this BD in internal memory to know where to put a packet that is received from the network.

When a packet is received from the network, a BD gets updated with information regarding the received packet and the packet is DMAed to a location in host memory described by the BD. The BCM57XX family (as the producer) then updates the Receive Return Ring Producer Index in the Status Block register corresponding to one of host memory's Receive Return Rings, and DMAs the BD to that Receive Return Ring.

It is the responsibility of the host software to setup, initialize, and manage the data structures in host memory, namely the Receive Producer Rings and the Receive Return Rings. The Producer/Consumer indices in the Mailbox and Status Block are read and updated by the host and BCM57XX family for this purpose.

RECEIVING JUMBO FRAMES

The BCM5700, BCM5701, BCM5702, BCM5703C, BCM5703S, BCM5704C, and BCM5704S MACs support Jumbo Ring for supporting Jumbo Frames. These devices also support the Extended Rx BDs for supporting the use of up to 4 host buffers for receiving one packet. The Jumbo Frames can be supported on these devices with or without the use of Extended Rx BDs. In order to support Jumbo Frames without Extended Rx BDs, the Jumbo Ring should be programmed with the Rx BDs each pointing to large enough host buffer for receiving a Jumbo Frame. This method requires the host to allocate large buffers of contiguous memory. The use of Extend Rx BDs in Jumbo Ring allows chaining of up to four host buffers for receiving a Jumbo Frame and hence the receive buffers need not be large buffers of contiguous memory. The RCB_FLAG_USE_EXT_RECV_BD flag of Jumbo Ring RCB should be set to 1 for using the Extended BDs.

The BCM5705, BCM5788, BCM5721, BCM5751, and BCM5752 devices do not support Jumbo Frames and hence both Jumbo Ring and Extended Rx BDs are not supported.

The BCM5714C, BCM5714S, BCM5715C, and BCM5715S devices support only one send ring, one standard receive producer ring, and one receive return ring. These devices also support Extended Rx BDs for supporting chaining of up to 4 host buffers for a received packet. The BCM5714C, BCM5714S, BCM5715C, and BCM5715S devices also support Jumbo Frames. In order to support Jumbo Frames without Extended Rx BDs, the Standard Receive Ring should be programmed with the Rx BDs each pointing to large enough host buffer for receiving a Jumbo Frame. This method requires the host to allocate a large buffer of contiguous memory for receiving even a smallest sized Ethernet packet. The use of Extended Rx BDs in Standard Ring allows chaining of up to four host buffers for receiving a Jumbo Frame and hence the receive buffers need not be large buffers of contiguous memory. The RCB_FLAG_USE_EXT_RECV_BD flag of Standard Ring RCB should be set to 1 for using the Extended BDs. Please note the following for enabling the Jumbo Frame Support of these devices with the use of Extended BDs.

1. Enable the Standard Ring to support Extended Rx BDs by setting the bit 29 of Write DMA Engine Mode register (offset 0x4C00). By default, the Standard Receive Producer Ring is enabled only fro normal Rx BDs and the bit 29 of WDMA Mode register is set to zero.
2. The Extended BD is 64 bytes long and hence the maximum number of Extended BDs supported by Standard Receive Producer Ring and Receive Return Ring is 256.
3. The Reserved field in Extended Rx BD (offset 0x1C bits 15:0; Please see “Extended Receive Buffer Descriptor” on page 122 for Extended BD format) is used by the RDI and Write DMA Engine for saving Length0. This is needed as the length0 field in Extended RBD (offset 0x28 bits 15:0) is modified by the RDI (Receive Data Initiator Engine) with the total length of the incoming packet.

Section 7: Transmit Data Flow

INTRODUCTION

Send Buffer Descriptors (BDs) begin on the Send Producer rings. The BCM5700 with external memory supports up to 16 Host Based or Controller Based Send Rings. The BCM5700 without external memory, BCM5701, BCM5702, BCM5703C, BCM5703S, BCM5704C, and BCM5704S MACs support up to 4 Host Based or Controller Based Send Rings. Multiple Send Rings can be utilized by host software to select varying levels of priority and thus support varying levels of quality of service. The BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714C, BCM5714S, BCM5715C, BCM5715S devices support only one Send Ring.

In BCM5700, enabling Send Producer Rings 5 to 16 displaces the internal memory buffer pool for Ethernet packets. The hardware application must make external memory available to provide packet buffer memory for RX/TX traffic.

Send Producer ring 1 has the highest transmit priority and Send Producer ring 16 has the lowest priority. The MAC selects Send BDs totally based on priority, and available BDs on the higher priority ring are always selected first. The scheduling of the BDs is priority-based and there is no fairness scheme to prevent starvation.

The host device driver updates the Mailbox to reflect available Send BDs.

- The MAC moves the available Send BDs to device local memory—a cache.
- Next, the MAC selects a BD from the internal cache using priority scheduling.

The physical address, programmed in the Send BD by the host device driver prior to the Mailbox update, contains the host memory location of the TX packet buffer. The MAC reads the address from Send BD and schedules a bus master DMA for reading the packet data from host buffer. The packet data will be moved into device internal buffers from host buffers by Read DMA engine, and all the read buffers of 1 packet are chained together into a cluster. This cluster is then sent to the transmit MAC which sends the packet data to the integrated PHY for transmission on Ethernet media.

The write DMA engine will subsequently update the status block to indicate that the Send BD was consumed. The host driver normally returns the packet buffers to the NOS/protocol so the next packet can reuse that host physical memory. The send BD is available for the next TX packet.

SEND RINGS

The send rings are shared data structures that are used to describe a series of data buffers that will be transferred onto the network. The shared data structure is called the Ring Control Block (RCB), and the entries within a ring for describing the data buffers are called the Send Buffer Descriptors (Send BDs).

Associated with each ring are two indices that control its operation. These indices are the producer index and the consumer index, which are not shared between the host software and the BCM57XX family. In the case of send rings, the host software controls the producer index by adding elements (initializing a Send BD) to the ring and incrementing the index. Similarly, the BCM57XX family controls the consumer index by removing elements (processing a Send BD) from the ring and incrementing the index.

The host software is responsible for maintaining its producer index and updating it by writing to the appropriate send ring host/NIC producer index mailbox register (starting at offset 0x200 through 0x3FF for host standard and flat modes and offset 0x5800 through 0x59FF for indirect mode). The mailbox registers are described in "Mailbox" on page 123, "High-Priority Mailboxes" on page 370 (for offsets 0x200 through 0x3FF), and "Low-Priority Mailboxes" on page 490 (for offsets 0x5800 through 0x59FF). The update actually triggers the BCM57XX family to process the send descriptors starting at its consumer index. As a descriptor is processed, the consumer index is incremented, and the new index is reflected in a new status block update. Status block is described in "Status Block" on page 103.

When the producer and consumer indices are equal, the ring is empty. When the producer index is one behind the consumer, the ring is full. Because of this configuration, the producer index always points to an empty slot. Thus, there will always be at least one empty slot in a ring.

The BCM57XX family has two operating modes that determine the location of the send rings. With the NIC-based send ring mode, the rings reside in NIC memory (see "NIC-Based Send Ring" on page 138). With the host-based send ring mode, the rings are located in host memory (see "Host-Based Send Ring" on page 139).

Figure 54 illustrates the relationships between all the components of a send ring.

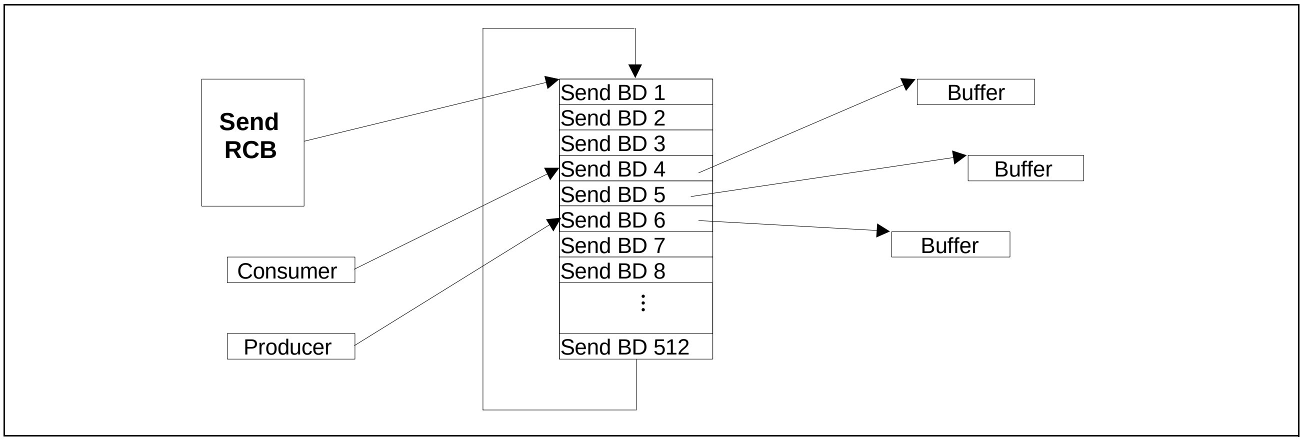
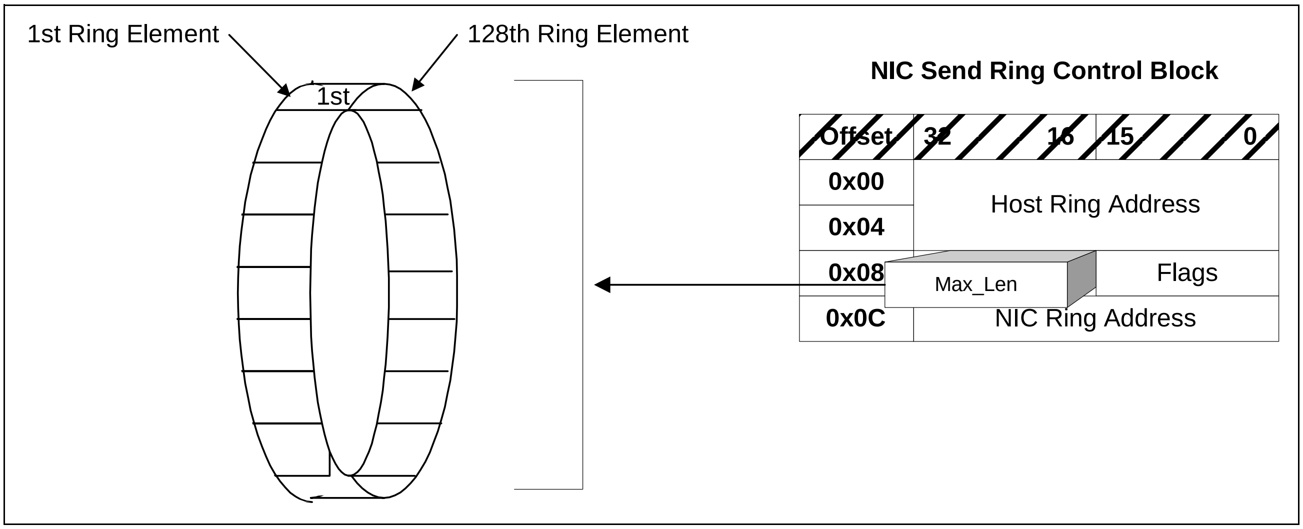


Figure 54: Relationships Between All Components of a Send Ring

Ring Control Block

Each send ring has a Ring Control Block (RCB) associated with it. The RCB contains a pointer to the first Send BD in the device and host memory, number of send BDs in the ring, and control flags (see "Send Rings" on page 92 for a full discussion of the send RCB). All the fields are in big-endian ordering as required by the BCM57XX family. The RCBs of the send rings are located in the NIC's Miscellaneous Memory Region at offset 0x0100.

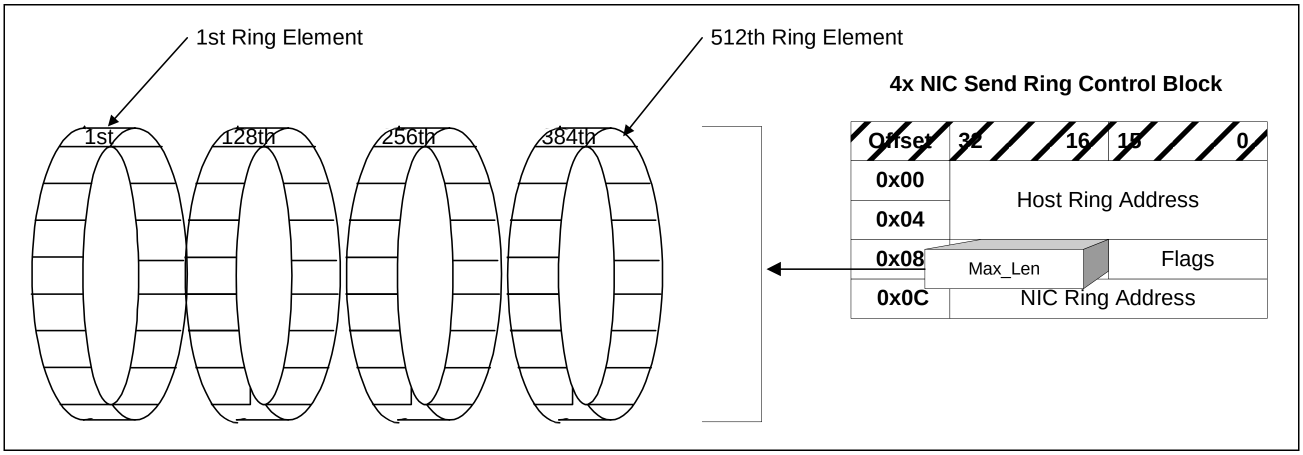
Send rings may reside in either host or NIC memory (The BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714C, BCM5714S, BCM5715C, and BCM5715S devices support only one host based Send Ring). When the ring is NIC-based, the entire ring data structure resides in the NIC local memory. A NIC send ring contains 128 send BDs by default (see Figure 55). The host device driver must configure the Max_Len field of the value of 128. The host driver must configure the NIC_Ring_Address to point a memory map region for the send BDs (see "Memory Maps and Pool Configuration" on page 171).



Offset	32 16	15 0
0x00	Host Ring Address	
0x04		
0x08	Max_Len	Flags
0x0C	NIC Ring Address	

Figure 55: NIC Send Ring

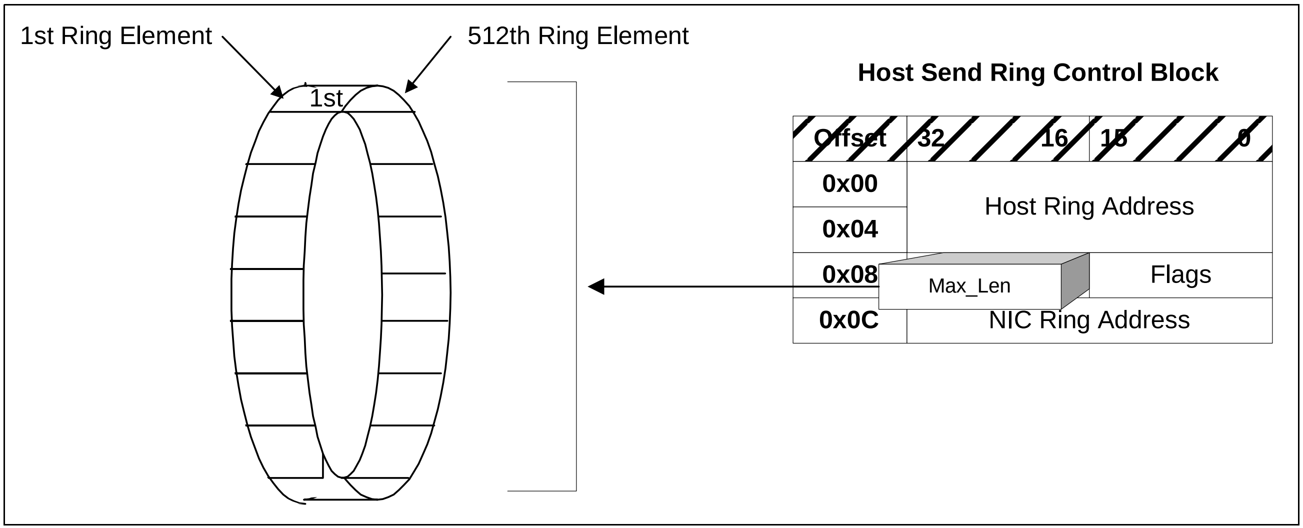
NIC local rings may be combined (see Figure 56). When the rings are combined, the total size of the ring may be increased to 512 entries. The total number of NIC based send rings decrease to one when the four adjacent NIC based send rings are combined. The host device driver should set the Max_Len field to 512 when the four NIC based send rings are combined.



Offset	32	16	15	0
0x00	Host Ring Address			
0x04				
0x08	Max_Len		Flags	
0x0C	NIC Ring Address			

Figure 56: Combining NIC Local Rings

The send rings may be completely located in host memory. In case of host based send rings, the send BDs will be bus-mastered from host memory into device local memory. The host device driver will program the BDs directly in its memory space and avoid programmed I/O to the MAC. The Max_Len field in the RCB (see Figure 57) should be set to the value of 512 for all host-based send rings.



Offset	32	16	15	0
0x00	Host Ring Address			
0x04				
0x08	Max_Len		Flags	
0x0C	NIC Ring Address			

Figure 57: Max_Len Field in Ring Control Block

NIC-BASED SEND RING

The send buffer descriptors of a send ring could reside in NIC memory. This mode of operation is referred to as NIC-based send ring. The host software configures the BCM57XX family to operate in this mode by clearing the Mode_Control.Host_Send_BDS bit at offset 0x6800 and the RCBs of the send rings configured accordingly as described in the "Ring Control Block" on page 136.

The BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714C, BCM5714S, BCM5715C, and BCM5715S devices do not support NIC based Send Rings.

Each NIC-based send ring has 128 buffer descriptors. The location of these descriptors is located in the NIC memory at offset 0x4000. This area is used as the staging area when the BCM57XX family is configured to operate in the host-based send ring mode.

In the NIC-based send ring mode, the host software initializes the buffer descriptors directly in the NIC memory, and then increments the producer index by the number of initialized descriptors. The new index is then written to the corresponding send ring NIC producer index mailbox register (starting at offset 0x380 for host standard and flat modes and offset 0x5980 for indirect mode—see "Send BD Ring 1-16 NIC Producer Indices Registers (Offset 0x380-0x3F8)" on page 374 and "Send BD Ring 1-16 NIC Producer Indices Registers (Offset 0x5980-0x59F8, BCM5700 and BCM5701 Only)" on page 493), which may trigger an event for the BCM57XX family to process the descriptors. Eventually, the data associated with descriptors are transferred onto the network. The BCM57XX family maintains the send ring consumer index, which is incremented as it processes the send ring buffer descriptors.

The BCM57XX family informs the host software of its progress by updating the consumer index in the status block. The host software uses the consumer index and its producer index to determine the empty slots in the send ring. The BCM57XX family implements an algorithm that periodically DMAs the status block to host memory in an efficient manner.

There are 16 (4 if there is no external memory) NIC-based send rings each containing 128 descriptors. The host software may extend the ring size to 512 descriptors by combining four adjacent send rings into one; however, the total ring count is reduced to four from sixteen when external SSRAM is present and to one from four when only device internal memory is used. The host device driver should assert the 4x_Size_NIC_Based_Send_Ring bit in the Mode Control register (see "Mode Control Register (Offset 0x6800)" on page 502) for extending the NIC based Send Ring Size. The rings are combined in the manner shown in Table 60.

Table 60: Combining Send Rings

Ring	*Offset*	*Description*
1	0x4000	Combine Rings 1 - 4
2	0x6000	Combine Rings 5 - 8 (Must have external memory)
3	0x8000	Combine Rings 9 - 12 (Must have external memory)
4	0xa000	Combine Rings 13 - 16 (Must have external memory)

HOST-BASED SEND RING

The send buffer descriptors of a send ring could reside in host memory. This mode of operation is referred to as host-based send ring. The host software configures the BCM57XX family to operate in this mode by setting the Mode_Control.Host_Send_BDs bit at offset 0x6800 and the RCBs of the send rings configured accordingly as described in "Ring Control Block" on page 136.

The BCM5700 with external memory supports up to 16 host based send rings. The BCM5700 with only internal memory, BCM5701, BCM5702, BCM5703C, BCM5703S, BCM5704C, and BCM5704S devices support up to 4 host based send rings. The BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714C, BCM5714S, BCM5715C, and BCM5715S devices support only one host based send ring.

Each host-based send ring has 512 buffer descriptors, which are periodically and transparently DMAed to a staging area inside the NIC's internal memory where they are waiting to be consumed. The staging area can hold up to 128 entries per-ring, and BCM57XX family will try to keep the staging area full at all times by constantly monitoring the consumer and producer index (the algorithm for accomplishing this is beyond the scope of this manual). The staging areas are located at a starting offset 0x4000 of NIC memory. Figure 58 illustrates the relationship between the send buffer descriptors in host memory and the staging area in NIC memory.

Whenever the host software initializes new buffer descriptors, its send ring producer index is incremented by the number of descriptors. The new index is then written to the corresponding send ring host producer index mailbox register (starting at offset 0x300 for host standard and flat modes and offset 0x5900 for indirect mode—see "Send BD Ring 1-16 Host Producer Indices Registers (Offset 0x300-0x378)" on page 374 and "Send BD Ring 1-4 Host Producer Indices Registers (Offset 0x5900-0x5918)" on page 493), which may trigger the BCM57XX family to DMA the descriptors to its staging area. Eventually, the buffer descriptors are processed, and the data associated with these descriptors is transferred onto the network.

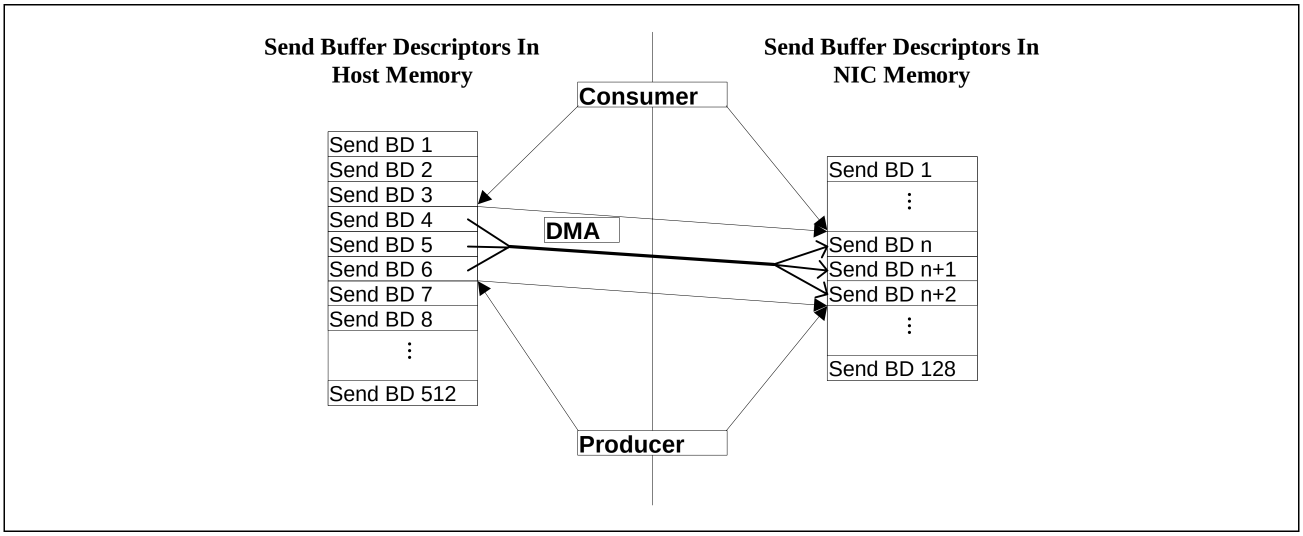


Figure 58: Relationship Between Send Buffer Descriptors

The BCM57XX family maintains the ring consumer index, which is incremented as it processes the descriptors. The BCM57XX family informs the host software of its progress by updating the consumer index in the status block. The host software uses the consumer index and its producer index to determine the empty slots in the ring. The BCM57XX family implements an algorithm that periodically DMAs the status block to host memory in an efficient manner.

CHECKSUM OFFLOAD

As network speed increases, offloading is becoming an important feature, and the ability to offload tasks from the host processor aids in the efficiency of the host and in overall system performance. To achieve a significant performance boost, most operating systems now a days offer a mechanism for the TCP/IP protocol stack to offload checksum calculations to the device.

The host software can configure the BCM57XX family to calculate IP, TCP, and UDP checksum as described in RFC 791, RFC 793, and RFC 768 respectively. The first step in checksum calculation is determining the start of an IP and UDP datagram and TCP segment within a frame, which could vary depending on whether the frame is tagged (VLAN) or encapsulated with LLC/SNAP header. Then the checksum is computed from the start to the end of the datagram and inserted into the appropriate location in protocol header. BCM57XX family is designed to support checksum calculation on all frame types and also on IP datagram and TCP segments containing options.

In order for the BCM57XX family to compute the checksum and insert it into the outgoing frame, the host software must set the appropriate control bits in the send buffer descriptors associated with the frame and seed the checksum field with zero or with the pseudo header checksum.

The host software enables IP checksum calculation by setting the IP_CHKSUM bits in all the send buffer descriptors associated with the frame. The BCM57XX family inserts the checksum into the checksum field of the IP header.

To enable TCP or UDP checksum calculation, the host software must set the TCP_UDP_CKSUM bit in all the send buffer descriptors associated with the frame containing the entire UDP datagram or TCP segment. However, if TCP segment is fragmented into IP fragments (each contained within a frame), all the send buffer descriptors of the IP fragments must also have the IP_FRAG bit set except for the last descriptor, which has the IP_FRAG_END bit set. Also, the descriptors must be consecutive and in the correct order. The UDP checksum engine does not span IP fragmented frames. The IP_FRAG and IP_FRAG_END flags do not enable UDP checksum capability when the IP layer has fragmented the UDP message.

The host software can configure the BCM57XX family to disable TCP or UDP pseudo-header checksum calculation by setting the Mode_Control.Send_No_Pseudo_Header_Checksum bit. When set, the host software must seed the checksum field in the TCP or UDP header with the pseudo-header checksum. If the Mode_Control.Send_No_Pseudo_Header_Checksum is cleared, the BCM57XX family will compute the checksum including the pseudo header and insert it into the checksum field.

SCATTER/GATHER

Most often, the host software requests the NIC to transmit a frame that spans several physical fragments that are arbitrary in size and buffer alignment. This requires the BCM57XX family to gather all these fragments during a DMA process into a continuous data stream for transmission. On the other hand, the BCM57XX family is also capable of scattering incoming frame into several physical fragments.

The ability to scatter/gather a frame lessens the restriction on the host software and increases overall system performance. For example, a TCP/IP protocol stack could preconstruct the MAC and IP headers in separate buffers that are combined with the payload to form a complete frame. Since the header data are fairly constant during a TCP or UDP session, the stack could use the same header buffers for the next frame.

The BCM57XX family uses a buffer descriptor for describing a physical fragment. There are two types of buffer descriptors; the Receive MAC processes receive buffer descriptors (Receive BD) and the Transmit MAC processes send buffer descriptors (Send BD).

Figure 59 illustrates the relationship between a frame consisting of multiple fragments and their corresponding send buffer descriptors.

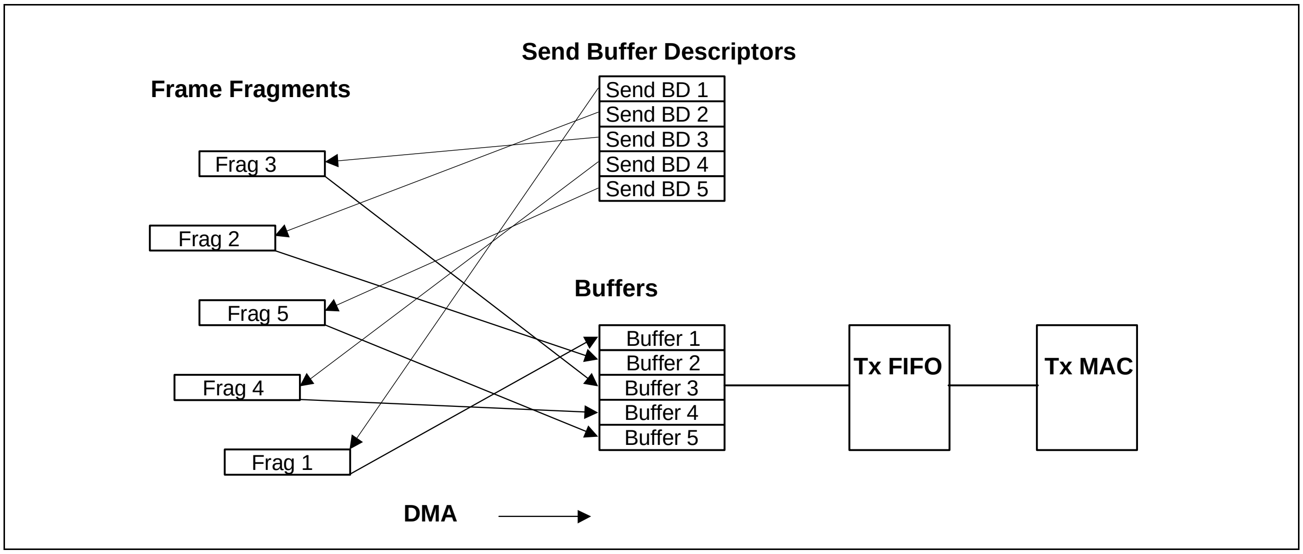


Figure 59: Scatter Gather of Frame Fragments

To transmit a frame, the host software sets up consecutive buffer descriptors in a send ring. Each buffer descriptor describes a physical fragment of a frame. As an example, the above figure illustrates a frame consisting of five fragments that are scattered throughout host memory. Frag1, the first fragment, is at the start of the frame, and Frag5, the last fragment, is at the end of a frame. For each fragment, there is a corresponding buffer descriptor, SendBd1 through SendBd5. These buffer descriptors must be initialized in the send ring in a consecutive order, SendBd1 to SendBd5. The last send buffer descriptor of a frame must have the PACKET_END bit of Send BD Flags field set to indicate the end of a frame.

VLAN Tag Insertion

The BCM57XX family is capable of inserting 802.1Q compliant VLAN tags into transmitted frames and extracting the tags from received frames. A frame containing the 802.1Q VLAN tag has the value TPID (Tag Protocol Identifier) value in the Ethertype field followed by a 16-bit TCI (Tag Control Information) field, which is made up of one CFI bit, 3 802.1P priority bits, and a 12-bit VLAN ID. The original 16-bit Ethertype/Length field follows the TCI field.

Table 59 on page 130 shows the frame format with 802.1Q VLAN tag inserted.

The BCM57XX family allows the host software to enable or disable tag insertion on a per-packet basis. To send a frame with a VLAN tag, the host software must initialize the first send buffer descriptor of a packet with the VLAN tag value and set the VLAN_TAG bit of Send BD Flags field (see "Send Rings" on page 135).

TX Data Flow Diagram

Figure 60 on page 143 illustrates how a frame, consisting of several fragments, is sent from the host to the NIC and onto the network. For simplicity, the diagram depicts the operation of a single ring.

1. The host software calls a system API to retrieve the three physical fragments of the frame. It initializes the next three send buffer descriptors to point to each fragment. The send buffer descriptors reside in host memory or NIC memory depending on how the send rings are configured (see "Send Rings" on page 135). Internally, the host software maintains the ring's producer index. In this case, the producer index is incremented by three because there are three fragments.
2. The host software updates the send producer index by writing the value to ring's Send Producer Index Mailbox at offset 0x300 for host standard and flat modes and offset 0x5900 for indirect mode. The update triggers the BCM57XX family to process the send buffer descriptors.
3. If the BCM57XX family is operating the host-based send ring mode, three send buffer descriptors are DMAed to the ring's staging area in NIC memory as indicated in the RCB.
4. The BMC5700 DMAs the frame (as described in the descriptors) to its internal memory for transmission.
5. Internally, the BCM57XX family maintains the ring's consumer index, which is incremented as it processes the descriptors.
6. The new consumer index is written to the status block in NIC memory (see "Status Block" on page 103).

7 The status block is DMAed to host memory. This DMA is subject to Host Coalescing, and the NIC may generate an interrupt at this point.

The following figure shows the basic driver flow to send a packet.

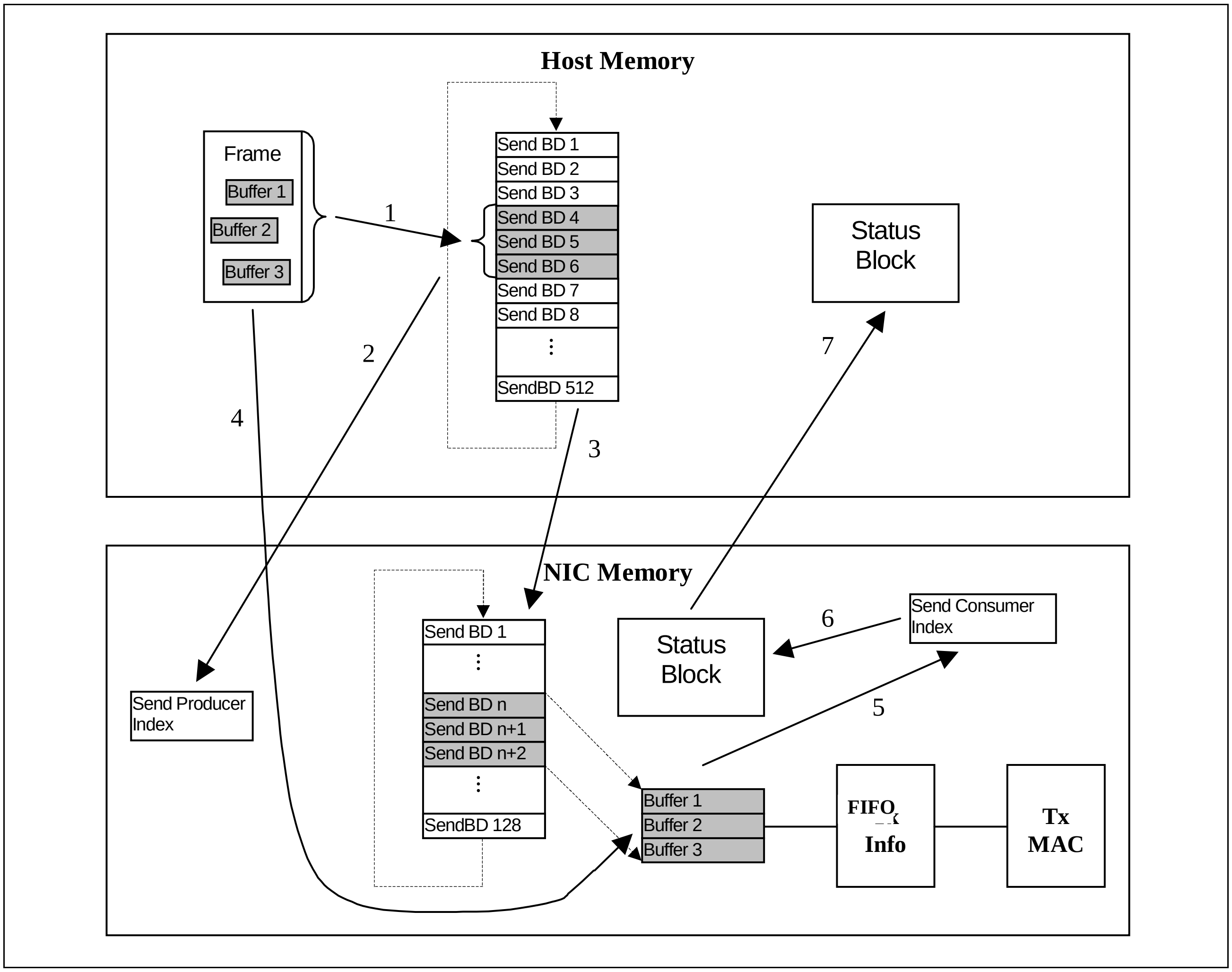


Figure 60: Transmit Data Flow

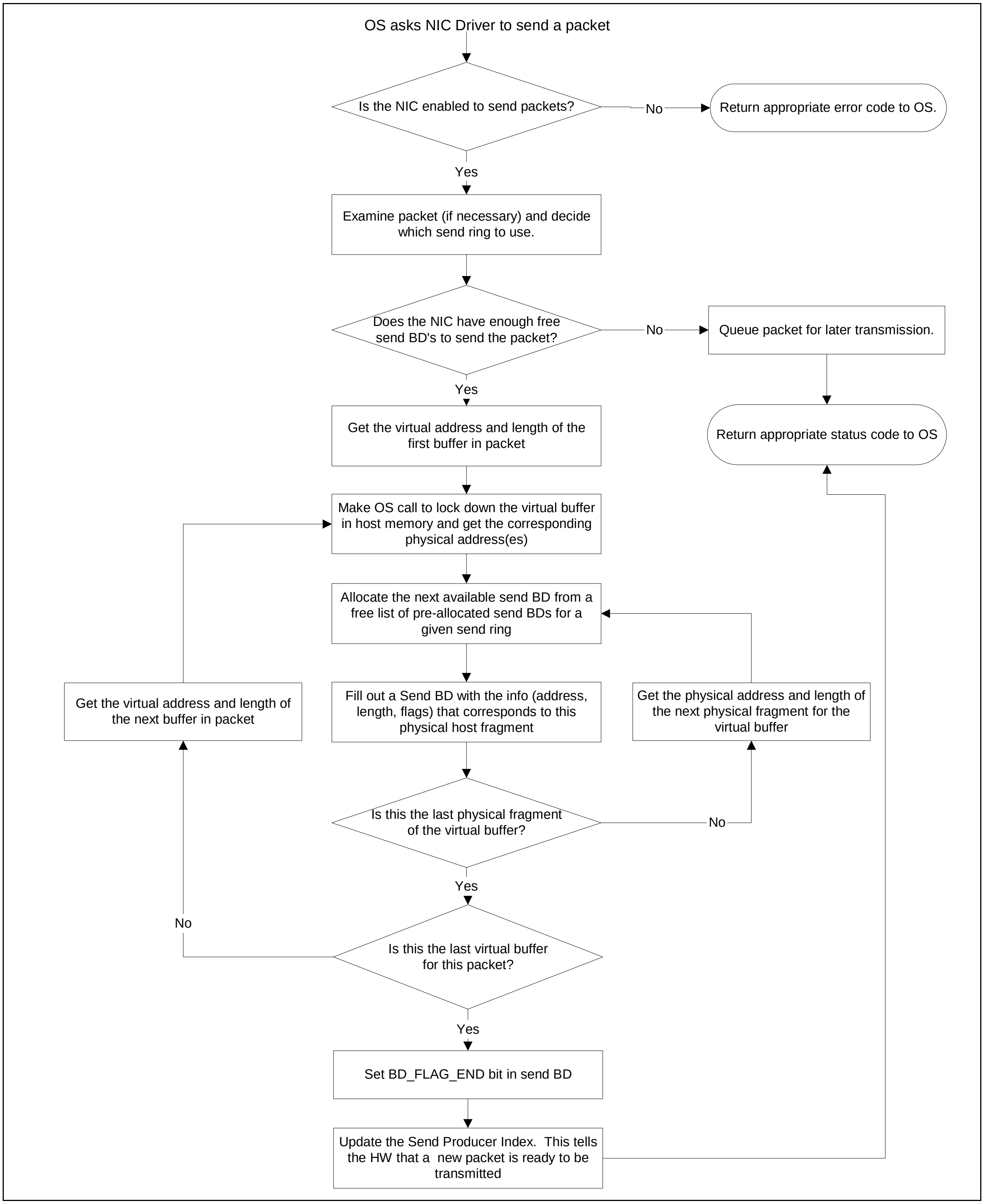

OS asks NIC Driver to send a packet
Is the NIC enabled to send packets?
No
Return appropriate error code to OS.
Yes
Examine packet (if necessary) and decide which send ring to use.
Does the NIC have enough free send BD's to send the packet?
No
Queue packet for later transmission.
Yes
Return appropriate status code to OS
Get the virtual address and length of the first buffer in packet
Make OS call to lock down the virtual buffer in host memory and get the corresponding physical address(es)
Allocate the next available send BD from a free list of pre-allocated send BDs for a given send ring
Get the virtual address and length of the next buffer in packet
Fill out a Send BD with the info (address, length, flags) that corresponds to this physical host fragment
Get the physical address and length of the next physical fragment for the virtual buffer
Is this the last physical fragment of the virtual buffer?
No
Yes
No
Is this the last virtual buffer for this packet?
Yes
Set BD_FLAG_END bit in send BD
Update the Send Producer Index. This tells the HW that a new packet is ready to be transmitted

Figure 61: Basic Driver Flow to Send a Packet

TRANSMITTING JUMBO FRAMES

The mechanism to transmit a Jumbo Frame in BCM57XX NetXtreme devices is exactly same as transmitting a normal Ethernet Frame as described in "TX Data Flow Diagram" on page 142. The BCM5700, BCM5701, BCM5702, BCM5703C, BCM5703S, BCM5704C, BCM5704S, BCM5714C, BCM5714S, BCM5715C, and BCM5715S support the Jumbo Frames.

In BCM5714C, BCM5714S, BCM5715C, and BCM5715S devices, the following hardware modifications have been made in the Transmit data path to be able to store up to two jumbo frames in the Transmit MBUF memory.

Increased the TX MBUF to 22 KB (176 MBUFs). This total of 176 MBUFs can be configured as:

- 173 MBUFs for TX packet + 3 MBUFs for IPMI messages
- 163 MBUFs for TX packet + 13 MBUFs for IPMI messages
- 158 MBUFs for TX packet + 18 MBUFs for IPMI messages
- 141 MBUFs for TX packet + 35 MBUFs for IPMI messages

Note: The BCM5714C, BCM5714S, BCM5715C, and BCM5715S devices cannot transmit Jumbo Frames when Large Send Offload (LSO) feature of these devices is enabled. This is because the LSO engine will always breakup up the large TCP segment into a normal sized Ethernet frames for transmission on Ethernet media. Also, note that the CPU cannot transmit a Jumbo Frame using the MBUFs reserved for ASF/IPMI messages, since only a maximum of 35 MBUFs can be reserved for ASF/IPMI messages.

Section 8: Device Control

INITIALIZATION

DESCRIPTION

This section provides programmers a procedure for initializing the NetXtreme family of devices. There is a specific sequence of steps that must be taken to enable this device. This section assumes the host programmer can allocate physical memory for various control blocks using OS/RTOS specific methods. Ring Control Blocks (RCBs) and Buffer Descriptors (BDs) all require host physical addresses; the MAC uses bus-master DMA to move packet data to host memory. The methods for allocation and de-allocation of host physical memory are beyond this document's scope.

INITIALIZATION PROCEDURE

This section lists the initializing procedure for the MAC portion of the NetXtreme family of devices.

1. Enable MAC memory space decode and bus mastering (optional). If the device has not been initialized previously (power on reset), the host software must enable these bits to issue the core clock reset in Step 7. Set the Bus_Master and Memory_Space bits in the PCI Configuration Space Command register (see "Command Register (Offset 0x04)" on page 302).

2. Disable interrupts (optional). If the device has not been initialized previously (power-on reset), the host software should disable and clear interrupts prior to the core_clock reset in Step 7. Set the Mask_PCI_Interrupt_Output and Clear_Interrupt_INTA bits in the Miscellaneous Host Control register (see "Miscellaneous Host Control Register (Offset 0x68)" on page 325).

3. Save the PCI Cache Line Size register and PCI Subsystem Vendor ID registers (see "Cache Line Size Register (Offset 0x0C)" on page 305 and "Subsystem ID Register (Offset 0x2E)" on page 309) in the PCI configuration space to temporary variables. These registers must be restored after a core clock reset in Step 7.

4. Acquire the NVRAM lock for BCM5702 and later devices by setting the REQ_SET1 bit of the Software Arbitration register (see "Software Arbitration Register (Offset 0x7020)" on page 555) and then waiting for the ARB_WON1 bit to be set.

5 Prepare the chip for writing T3_MAGIC_NUMBER to device memory location 0xB50.

a. Set the Enable bit in the Memory Arbiter Mode register (see "Memory Arbiter Mode Register (Offset 0x4000)" on page 460).

b. Set the Enable_Indirect_Access bit of the Miscellaneous Host Control register (see "Miscellaneous Host Control Register (Offset 0x68)" on page 325).

c. Initialize the Endian_Word_Swap and Endian_Byte_Swap bits of Miscellaneous Host Control register (offset 0x68) depending on the Host Platform Endianness (This step is required only if Indirect Access registers are accessed through memory mapped accesses).

d. Initialize the Byte_Swap_Non_Frame_Data and Word_Swap_Non_Frame_Data bits of Mode Control register (offset 0x6800) to the required values depending on the Host Platform Endianness.

6. Write the T3_MAGIC_NUMBER (i.e., 0x4B657654) to the MAC memory at 0xb50 to notify the bootcode that the following reset is a warm reset.

7. Reset the core clocks. Set the CORE_CLock_Blocks-Reset bit in the General Control Miscellaneous Configuration register (see "Miscellaneous Configuration Register (Offset 0x6804)" on page 504). The GPHY_Power_Down_Override bit (bit-26) should also be set for the following MACs:
   - BCM5705
   - BCM5788
   - BCM5721
   - BCM5751
   - BCM5752
   - BCM5714C
   - BCM5715C
   - BCM5715S

   The Disable_GRC_Reset_on_PCI-E_Block bit (bit-29) should also be set for the following PCIe MACs:
   - BCM5721
   - BCM5751
   - BCM5752
8. Wait for core-clock reset to complete. Software should wait 100 µs for PCI and PCI-X systems, and 100 ms for PCIe systems. The core clock reset will disable indirect mode and flat/standard modes—software cannot poll the core-clock reset bit to de-assert, since the local memory interface is disabled by the reset.
9. Disable interrupts. Set the Mask_PCI_Interrupt_Output bit in the Miscellaneous Host Control register (see "Miscellaneous Host Control Register (Offset 0x68)" on page 325). The bit was reset after the core_clock reset, and interrupts must be masked off again.
10. Enable MAC memory space decode and bus mastering. Set the Bus_Master and Memory_Space bits in the PCI Configuration Space Command register (see "Command Register (Offset 0x04)" on page 302).
11. Disable PCI-X Relaxed Ordering. Clear the Enable_Relax order bit in the PCI-X Command register (see "Command Register (Offset 0x04)" on page 302).
12. Enable the MAC memory arbiter. Set the Enable bit in the Memory Arbiter Mode register (see "Memory Arbiter Mode Register (Offset 0x4000)" on page 460). Make sure that no other bits of this register are modified in case of BCM5714C, BCM5714S, BCM5715C, and BCM5715S MACs as these controllers use bits 31-30 of this register for configuration of Number of TxMbufs allocated for on-chip RISC processor.
13. Enable External Memory (optional). This step is only necessary if your application uses external SSRAM (BCM5700 MAC only). Write the external_memory_enable, size of extra memory and, external_memory_bank_select bits to the Miscellaneous Local Control register (see "Miscellaneous Local Control Register (Offset 0x6808)" on page 507).

    Wait 10 ms for external memory to initialize (optional). Polling is not required.
14. Initialize the Miscellaneous Host Control register (see "Miscellaneous Host Control Register (Offset 0x68)" on page 325):

    a. Set Endian Word Swap (optional). When the host processor architecture is big endian, the MAC may wordswap data, when acting as a PCI target device. Set the Enable_Endian_Word_Swap bit in the Miscellaneous Host Control register.

    b. Set Endian Byte Swap (optional). When the host processor architecture is big-endian, the MAC may byteswap data, when acting as a PCI target device. Set the Enable_Endian_Byte_Swap bit in the Miscellaneous Host Control register.

    c. Enable the indirect register pairs (see "Indirect Mode" on page 185). Set the Enable_Indirect_Access bit in the Miscellaneous Host Control register.

    d. Enable the PCI State register to allow the device driver read/write access by setting the Enable_PCI_State_Register bit in the Miscellaneous Host Control register

    e. Enable the PCI Clock Control register (see "PCI Clock Control Register (Offset 0x74)" on page 334) to allow the device driver read/write access by setting the Enable_Clock_Control_Register bit in the Miscellaneous Host Control register (see "Miscellaneous Host Control Register (Offset 0x68)" on page 325).

15. Set Byte_Swap_Non_Frame_Data and Byte_Swap_Data in the General Mode Control register (see "Mode Control Register (Offset 0x6800)" on page 502).

16. Set Word_Swap_Data and Word_Swap_Non_Frame_Data (Optional). When the host processor architecture is little-endian, set these additional bits in the General Mode Control register (see "Mode Control Register (Offset 0x6800)" on page 502).

17. Poll for bootcode completion (optional for embedded applications that will not use the Broadcom bootcode firmware). The device driver should poll the general communication memory at 0xB50 (see Table 82 on page 171 in "Memory Maps and Pool Configuration" on page 171) for the one's complement of the T3_MAGIC_NUMBER (i.e., 0xB49A89AB). The bootcode should complete initialization within 1000 ms for Flash devices and 10000 ms for SEEPROM devices.

18 Initialize the Ethernet MAC Mode register (see "Ethernet MAC Mode Register (Offset 0x400)" on page 377). Write the value 0x00000000 to this register for all Copper controllers and write the value 0x0000000C for BCM5703S and BCM5704S SerDes Fiber controllers. The BCM5714S and BCM5715S use the GMII to talk to the integrated 1000-BASE-X PHY and hence write 0x00000000 to MAC Mode register for these controllers also.

Note: This register is not cleared by the core clock reset above.

19. Enable the PCIe bug fix for BCM5721, BCM5751, and BCM5752 MACs by setting the bits 25 and 29 of the TLP control register at offset 0x7C00 without modifying the other bits of this register.

20. Enable data FIFO protection for BCM5721, BCM5751, and BCM5752 PCIe MACs by setting the DATA_FIFO_Protect bit of the TLP Control register (see "TLP Error Counter Register (Offset 0x7D40)" on page 592).

21. Enable the hardware fixes for the BCM5704 B0 and later versions by setting the bits 10, 12, and 13 of the Hardware Fix register at offset 0x66.

22. Enable Tagged Status Mode (optional) by setting the Enable_Tagged_Status_Mode bit of the Miscellaneous Host Control register (see "Miscellaneous Host Control Register (Offset 0x68)" on page 325). (For further information on Tagged Status Mode see "Interrupt Processing" on page 287).

23. Restore the PCI Cache Line Size and PCI Subsystem Vendor ID registers (see "Cache Line Size Register (Offset 0x0C)" on page 305 and "Subsystem Vendor ID Register (Offset 0x2C)" on page 307) in the PCI configuration space. These registers were cleared by the core clock reset.

24. Clear the MAC statistics block for by writing zeros to the BAR+0x300 to BAR+0xB00 to clear the statistics block in MAC local memory. This step is not required for BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714C, BCM5714S, BCM5715C, and BCM5715S MACs as these devices support the statistics in register space (Offset 0x800-0x8FF) instead of device memory space.

25. Clear the driver statistics memory region. Write zeros to the host memory region where the statistics block will be DMA'd (see "Statistics Block" on page 107). This step is not required for BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714C, BCM5714S, BCM5715C, and BCM5715S MACs as these devices do not support the DMA of device statistics to host memory at regular intervals.

26. Clear the driver status memory region. Write zeros to the host memory region where the status block will be DMAed (see "Statistics Block" on page 107).

27. Set the default PCI Command Encoding for Read/Write Transactions (see "PCI Command Usage" on page 209). The Default_PCI_Write_Command and Default_PCI_Read_Command bits in the DMA Read/Write Control register (see "DMA Read/Write Control Register (Offset 0x6C)" on page 327) must be initialized since they are zeroed after a device reset. Also initialize the DMA read and write watermarks.

Table 61: Recommended BCM57XX Setting for the DMA Read/Write Control Register

Broadcom MAC	Recommended Value
BCM5700, BCM5701	0x763F000F in PCI Bus Mode (i.e., if bit-2 of 0x70 is 0) 0x761B000F in PCI-X Bus Mode (i.e., if bit-2 of 0x70 is 1)
BCM5702	0x763F000F
BCM5703C, BCM5703S, BCM5704C, and BCM5704S	0x763F0000 in PCI bus mode (i.e., if bit-2 of 0x70 is 0) 0x769F0000 in PCI-X 33-MHz/50-MHz/66-MHz bus modes (i.e., if bit 2 of 0x70 is 1, and bits 4–0 of 0x74 is 0, 2, or 4) 0x769F4000 in PCI-X 100-MHz/133-MHz bus modes (i.e., if bit-2 of 0x70 is 1, and bits4–0 of 0x74 is 6 or 7)
BCM5721, BCM5751, and BCM5752	0x76180000 if the MaxPayLoadSize is 128 (i.e., bits 7–5 of 0xD8 = 000) 0x76380000 if the MaxPayLoadSize is 256 (i.e., bits 7–5 of 0xD8 = 001)
BCM5705, BCM5788	0x763F0000
BCM5714C, BCM5714S, BCM5715C, and BCM5715S	0x76144000

28. Set DMA byte swapping (optional). If the host processor architecture is big-endian, the MAC may byte swap both control and frame data, when acting as a PCI DMA master. Set the Byte_Swap_Non-Frame_Data, Byte_Swap_Data and Word_Swap_Data bits in the General Mode Control register (see "Mode Control Register (Offset 0x6800)" on page 502).

29. Configure the host-based send rings. If the device driver intends to keep the send ring(s) in host local storage rather MAC memory, the Host_Send_BDs bit in the General Mode Control register (see "Mode Control Register (Offset 0x6800)" on page 502) should be set. The Host_Send_BDs bit should be set for BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714C, BCM5714S, BCM5715C, and BCM5715S MACs as these devices support only one Send Ring that is host based.

30. Indicate Driver is ready to RX traffic. Set the Host_Stack_Up bit in the General Mode Control register (see "Mode Control Register (Offset 0x6800)" on page 502).

Note: Host software should be careful not to set the Route_Multicast_Frames_to_RISC core bit unless custom firmware has been developed for multicast frame handling. If this bit is set inadvertently, transmit frames (e.g., arp, broadcast) may be routed to the RISC Core and the data path will stall. Host software will observe BD and Frame buffers continue to DMA, but frames will not go out to the wire. Eventually, the MAC will run out of internal memory and both RX/TX will stall.

31. Configure TCP/UDP pseudo header checksum offloading. This step is relevant when TCP/UDP checksum calculations are offloaded to the device. The device driver may optionally disable receive and transmit pseudo header checksum calculations by the device by setting the Receive_No_PseudoHeader_Checksum and Send_No_PseudoHeader_Checksum bits in the General Mode Control register (see "Mode Control Register (Offset 0x6800)" on page 502). If the Send_No_PsuedoHeader_Checksum bit is set, the host software should make sure of seeding the correct pseudo header checksum value in TCP/UDP checksum field. Similarly if the Receive_No_PsuedoHeader_Checksum bit is set, the device driver should calculate the pseudo header checksum and add it to the TCP/UDP checksum field of the received packet.

Note: Some 57XX family adapters may calculate an incorrect pseudo-header checksum if the Send_No_PseudoHeader_Checksum is left at the default value of 0. Check the latest chip errata for affected chips and suggested workarounds. Broadcom's drivers will always set this bit.

32. Configure the frequency of MAC's free running 32-bit timer at offset 0x680C. The Timer_Prescaler bit field in the General Control Miscellaneous Configuration register (see "Miscellaneous Configuration Register (Offset 0x6804)" on page 504) sets the local frequency of this timer. This timer at 0x680C offset increments once every Timer_Prescalar number of core clock cycles. The core clock of the device runs at 66MHz and hence the device driver software should configure the Timer_Prescalar field with 0x41, or 65 decimal for the 0x680C timer to increment by one every 1 uS.

33 Configure MAC local memory pool. The MAC uses device local memory to buffer packets that will be DMAed to/from host memory. Host software needs to program the pool address differently based on the capabilities of the device. For example, external SSRAM will influence the pool location and size of BCM5700 device. The Mbuf Pool Base Address and Mbuf pool length registers (see "Memory Maps and Pool Configuration" on page 171) must be configured during initialization. Table 62 and Table 63 show the recommended configurations for the BCM57XX internal and BCM5700 external memory pool settings.

Table 62: Recommended BCM57XX Internal Memory-Only Memory Pool Settings

Register	Bits	Recommended Value
Mbuf Pool Base Address	All	0x8000 for BCM5700, BCM5701, BCM5702, and BCM5703 0x10000 for BCM5704C andBCM5704S Do not overwrite the bootcode settings for BCM5705, BCM5788, BCM5721, BCM5751, BCM5714C, BCM5714S, BCM5715C, BCM5715S, and BCM5752
Mbuf Pool Length	ALL	0x18000 0x10000 for BCM5704C and BCM5704S Do not overwrite the bootcode settings for BCM5705, BCM5788, BCM5721, BCM5751, BCM5714C, BCM5714S, BCM5715C, BCM5715S, and BCM5752

Note: In the BCM5705, BCM5788, BCM5721, BCM5751, BCM5714C, BCM5714S, BCM5715C, BCM5715S, and BCM5752 MAC-Transceivers, the default initialization values after reset for the Mbuf Pool Base Address is 0x10000 and the Mbuf Pool Length is 0x8000. The RISC scratchpad memory needs to be reserved out of the Rx Mbuf for these MACs and hence the bootcode calculates and sets the Mbuf Pool Base Address and Mbuf Pool Length registers of these MACs. So, Broadcom recommends to not to change these Mbuf Pool registers from driver software unless the driver software is loading a special firmware into the RISC scratch pad memory that needs to be reserved out of the Rx Mbuf memory. For designs without NVRAM (and hence bootcode), Broadcom recommends that the host driver use the default values of Mbuf Pool registers rather than setting these registers. If it is required to modify either of the Mbuf Pool registers, then driver software should clear the contents of RxMbuf memory and set the Reset RxMbuf Pointer bit of the Buffer Manager Mode register (offset 0x4400)

Note: For external memory use with the BCM5700 MAC, set Enable_External_memory and SRAM_size in GRC Miscellaneous Local Control register (see "Miscellaneous Local Control Register (Offset 0x6808)" on page 507) before setting Mbuf Pool Base Address and Mbuf Pool Length (see Table 63).

Table 63: Recommended BCM5700 External SSRAM Memory Pool Settings

Register	Bits	Recommended Value
Mbuf Pool Base Address	All	0x20000
Mbuf Pool Length	All	Pool Size Dependent on Hardware Application

34. Configure MAC DMA resource pool (for BCM5700/5701/5702/5703/5704 only). The BCM5700, BCM5701, BCM5702, 5703C, 5703S, and 5704 MACs use device local memory for DMA resources. These resources are necessary for bus master DMA to the host memory space. Host software must configure the DMA Pool Base Address and DMA Pool Length registers (see "Buffer Manager Control Registers" on page 466) during initialization (see Table 64). The device driver must allocate 8K for DMA resources.

Table 64: Recommended BCM57XX DMA Resource Pool Settings

Register	Bits	Recommended Value
DMA Descriptor Pool Base Address	All	0x2000
DMA Descriptor Pool Length	All	0x2000

35. Configure MAC memory pool watermarks. Broadcom has run hardware simulations on the Mbuf usage and strongly recommends the settings shown in Table 65. These settings/values will establish proper operation for 10/100/1000 speeds. Host software must configure the Read DMA Mbuf Low Watermark, MAC RX Mbuf Low Watermark, and Mbuf High Watermark registers (see "Read DMA MBUF Low Watermark Register (Offset 0x4410)" on page 469, "MAC RX MBUF Low Watermark Register (Offset 0x4414)" on page 470, and "MBUF High Watermark Register (Offset 0x4418)" on page 470) during initialization.

Table 65: Recommended BCM57XX MAC Memory Pool Watermark Settings

Register	Mini Frames	Standard Frames (BCM5700/5701/ 5702/5703/5704)	Standard Frames (BCM5705/5788/ 5721/5751/5714/ 5715/5752)	Jumbo Frames (BCM 5700/ 5701/5702/ 5703/5704)	Jumbo Frames (5714C/5714S/ 5715C/5715S)
Read DMA Mbuf Low Watermark	0x0	0x50	0x00	0x130	0x00
MAC RX Mbuf Low Watermark	0x10	0x20	0x10	0x98	0x4B
Mbuf High Watermark	0x60	0x60	0x60	0x17C	0x96

Note: The Low WaterMark Max Receive Frames register (0x504) specifies the number of good frames to receive after RxMbuf Low Watermark has been reached. The driver software should make sure that the MAC RxMbuf Low WaterMark is greater than the number of Mbufs required for receiving the number of frames as specified in 0x504. The first Mbuf in the Mbuf chain of a frame will have 80 bytes of packet data while each of the subsequent Mbufs except the last Mbuf will have 120 bytes for packet data. The last Mbuf in the chain will have the rest of the packet data which can be up to 120 bytes.

36. Configure DMA resource watermarks. Broadcom has run hardware simulations on the DMA resource usage and the recommendations shown in Table 66 are strongly encouraged. These settings/values will establish proper operation for 10/100/1000 speeds. Host software must configure the DMA Descriptor Low Watermark and DMA Descriptor High Watermark registers (see "DMA Descriptor Pool Low Watermark Register (Offset 0x4434)" on page 473 and "DMA Descriptor Pool High Watermark Register (Offset 0x4438)" on page 473) during initialization.

Table 66: Recommended BCM57XX DMA Memory Pool Watermark Settings

Register	Bits	Recommended Value
DMA Descriptor Low Watermark	All	5
DMA Descriptor High Watermark	All	10

37. Configure flow control behavior when the low watermark level has been reached (see Table 67 and "Low Watermark Maximum Receive Frames Register (Offset 0x504)" on page 394). Please see the note above in Step-35.

Table 67: Recommended BCM57XX Low Watermark Maximum Receive Frames Settings

Register	***Bits***	***Recommended Value***
Low Water Mark Maximum Receive Frames	All	2

38. Enable the buffer manager. The buffer manager handles the internal allocation of memory resources for send and receive traffic. The Enable and Attn_Enable bits should be set in the Buffer Manager Mode register (see "Buffer Manager Mode Register (Offset 0x4400)" on page 467).

39 Poll for successful start of buffer manager. Poll the Enable bit in the Buffer Manager Mode register (see "Buffer Manager Mode Register (Offset 0x4400)" on page 467) for 10 ms. This test ensures the buffer manager successfully starts from the previous step. The Enable bit will remain de-asserted until the buffer manager starts, at which point it will reflect an asserted state.

40. Enable internal hardware queues. The MAC architecture uses internal queues to pass messages between functional blocks. These messages coordinate RX/TX traffic flows. Device drivers need to enable these queues so the hardware blocks can pass messages. Host software must set and then reset the bits in the FTQ Reset register (see "FTQ Reset Register (Offset 0x5C00)" on page 494) to start internal queues:

a. First, host software should write 0xFFFFFFFF to the FTQ Reset register.

b. Second, host software should clear the FTQ Reset register by writing 0x00000000.

41. Initialize the Standard Receive Buffer Ring. Host software should write the Ring Control Block structure (see "Ring Control Blocks" on page 91) to the Standard Receive BD Ring RCB register (see "Standard Receive BD Ring RCB Register (Offset 0x2450)" on page 439). Host software should be careful to initialize the host physical memory address based on allocation routines specific to the OS/RTOS. Table 68 and Table 69 show the recommended Standard Ring Initialization settings.

Table 68: Recommended BCM57XX Standard Ring Initialization Settings for Internal Memory Only

RCB Data Field	***Recommended Value***	***Notes***
NIC Ring Address(32-bits)	0x6000	
Max_Length	0x600 (for BCM5700/5701/5702/5703C/5703S/5704C/5704S)	Max size Enet Frame + VLAN tag
	0x200 (for BCM5705/5788/5721/5751/5752)	Number of Elements in the ring
	0x200 (for 5714C/5714S/5715C/5715S when Standard Rx BDs are used)	Valid maximum value is 0x200
	0x100 (for 5714C/5714S/5715C/5715S when Extended Rx BDs are used)	Number of Elements in the ring

Table 69: Recommended BCM5700 Standard Ring Initialization Settings For External SSRAM

RCB Data Field	***Recommended Value***	***Notes***
NIC Ring Address (32-bits)	0xC000	
Max_Length	0x600	Max size Enet Frame + VLAN tag

Note: Host software must insure that on systems that support more than 4 GB of physical memory, Send Rings, Receive Return Rings, Producer Rings, and packet buffers are not allocated across the 4 GB memory boundary. For example, if the starting memory address of the Standard Receive Buffer Ring is below 4 GB and the ending address is above 4 GB, a Read DMA PCI Host Address Overflow error may be generated (see "Read DMA Status Register (Offset 0x4804)" on page 479).

42 Initialize the Jumbo Receive Buffer Ring (optional—BCM5700/5701/5702/5703C/5703S/5704C/5704S only). Host software should write the Ring Control Block structure (see "Ring Control Blocks" on page 91) to the Jumbo Receive BD Ring RCB register (see "Jumbo Receive BD Ring RCB Register (Offset 0x2440)" on page 438). Host software should be careful to initialize the host physical memory address based on allocation routines specific to the OS/RTOS. Table 70 and Table 71 show the recommended Jumbo Ring Initialization settings.

Table 70: Recommended BCM57XX Jumbo Ring Initialization Settings for Internal Memory Only

RCB Data Field	*Recommended Value*	*Notes*
NIC Ring Address (32-bits)	0x7000	
Flags	RCB_USE_EXT_RECV_BD	Extended Buffer Descriptors.
Flags	RCB_FLAG_RING_DISABLED	Default the ring disabled until initialization is complete.

Table 71: Recommended BCM5700 Jumbo Ring Initialization Settings for External SSRAM

RCB Data Field	*Recommended Value*	*Notes*
NIC Ring Address (32-bits)	0xD000	
Flags	RCB_USE_EXT_RECV_BD	Extended Buffer Descriptors.
Flags	RCB_FLAG_RING_DISABLED	Default the ring disabled until initialization is complete.

43. Initialize the Mini Receive Buffer Ring (Optional. BCM5700/5701/5702/5703C/5703S/5704C/5704S only). Mini Receive Producer Rings are only available to applications with external SSRAM. Host software should write the Ring Control Block structure (see "Ring Control Blocks" on page 91) to the Mini Receive BD Ring RCB register (see "Mini Receive BD Ring RCB Register (Offset 0x2460)" on page 439). Host software should be careful to initialize the host physical memory address based on allocation routines specific to the OS/RTOS. Table 72 shows the recommended Mini Ring Initialization settings.

Table 72: Recommended BCM5700 Mini Ring Initialization Settings for External SSRAM

RCB Data Field	*Recommended Value*	*Notes*
NIC Ring Address (32-bits)	0xE000	
Flags	RCB_FLAG_RING_DISABLED	Default the ring disabled until initialization is complete.

44. Set the BD Ring replenish thresholds for mini, standard, and jumbo RX Producer Rings. The threshold values indicate the number of buffer descriptors that must be indicated by the host software before a DMA is initiated to fetch additional receive descriptors in order to replenish used receive descriptors. Software should configure the following registers:

- Mini Receive BD Ring Replenish Threshold (see "Mini Receive BD Producer Ring Replenish Threshold Register (Offset 0x2C14)" on page 443)

Note: Only the BCM5700 MAC with external SSRAM can use mini rings.

- Standard Receive BD Ring Replenish Threshold (see "Standard Receive BD Producer Ring Replenish Threshold Register (Offset 0x2C18)" on page 443)
- Jumbo Receive BD Ring Replenish Threshold (see "Jumbo Receive BD Producer Ring Replenish Threshold Register (Offset 0x2C1C)" on page 443)

Broadcom suggests setting the RX Jumbo and RX Mini Producer rings to 1/8 of the total ring size. Table 73 shows some example replenish threshold settings based on the maximum number of BDs allocated.

Table 73: Examples of BCM57XX Replenish Threshold Settings

Ring	Total BDs	Threshold Value
Mini RX Producer	1024	128
Standard RX Producer	512	25
Jumbo RX Producer	256	16

Note: The Standard RX Producer threshold value should be set very low. Some O/S may run short of memory resource, and the number of BDs that are made available will decrease proportionally.

45. Disable unused send producer rings (BCM5700/5701/5702/5703C/5703S/5704C/5704S only). Host software should write the RCB flag bit RCB_FLAG_RING_DISABLE for all Send Rings that will not be utilized in the implementation. A maximum of four send rings are available, when the application implements device internal memory only (no external SSRAM). The BCM5700 can support up to 16 Send Rings when external SSRAM is used. The Send RCBs are located in the device Miscellaneous Memory region from 0x100 to 0x1FF.

46. Initialize send producer index registers in mailbox. Clear (i.e., zero) the Send BD Ring1-16 Host Producer Index and Send BD Ring1-16 NIC Producer Index registers (see "Send BD Ring 1-16 NIC Producer Indices Registers (Offset 0x380-0x3F8)" on page 374 and for host standard and flat modes). If host software is using register indirect mode, the Send BD ring 1-16 producer indices registers (see "Send BD Ring 1-4 Host Producer Indices Registers (Offset 0x5900-0x5918)" on page 493 and "Send BD Ring 1-16 NIC Producer Indices Registers (Offset 0x5980-0x59F8, BCM5700 and BCM5701 Only)" on page 493) may be cleared. It is unnecessary to clear both the high and low-priority mailbox registers.

47. Initialize send rings. The Send RCBs are located in the Miscellaneous Memory region from 0x100 to 0x1FF. Host software should be careful to initialize the host physical memory address, based on allocation routines specific to the OS/RTOS. The MAC will cache ¼ of the available Send BDs in NIC local memory, so the host driver must set up the NIC local address. The following formula should be used to calculate the NIC Send Ring address:

```
NIC Ring Address = 0x4000 + (Ring_Number * sizeof(Send_Buffer_Descriptor) *
NO_BDS_IN_RING) / 4
```

48. Disable unused Receive Return Rings. Host software should write the RCB_FLAG_RING_DISABLED bit to the flags field for each ring control block. There are 16 Receive Return rings in BCM5700/5701/5702/5703C/5703S/5704C/5704S MACs.

49. Initialize Receive Return Rings. The Receive Return RCBs are located in the Miscellaneous Memory region from 0x200 to 0x2FF. Host software should be careful to initialize the host physical memory address, based on allocation routines specific to the OS/RTOS. In BCM5700, the Max_len field of the RCB should be programmed to 2048 when Mini Receive Producer Ring is enabled. In BCM5700 without external SSRAM, BCM5701, BCM5702, BCM5703C, BCM5703S, BCM5704C, and BCM5704S devices, the Max_Len should be set to 1024. In BCM5705, BCM5788, BCM5721, BCM5751, and BCM5752 MACs, the Max_Len field should be set to 512. In BCM5714C/5714S/5715C/5715S devices, the Max_Len should be set to 512 or 256 depending on whether Standard Rx BDs are used or Extended Rx BDs are used.

50. The NIC RingAddress field of RCB is an invalid field for Rx Return Rings and hence host driver should set NIC Ring Address to 0x0000.

51 Initialize the Receive Producer Ring mailbox registers. The Jumbo and Mini rings are supported only in BCM5700/5701/5702/5703C/5703S/5704C/5704S MACs.

- Software should write the value 0x00000000 (clear) to the low 32 bits of the Receive BD Standard Producer Ring Index mailbox (see "Receive BD Standard Producer Ring Index Register (Offset 0x268)" on page 373 for host standard and flat modes). If software is using register indirect mode, the receive BD Standard Producer Ring index register (see "Receive BD Standard Producer Ring Index Register (Offset 0x5868)" on page 492) may be cleared. It is unnecessary to clear both the high- and low-priority mailbox registers.
- Software should write the value 0x00000000 to the low 32 bits of the Receive BD Jumbo Producer Ring Index mailbox (see "Receive BD Jumbo Producer Ring Index Register (Offset 0x270)" on page 373 for host standard and flat modes). If host software is using register indirect mode, the RX BD Jumbo Producer ring index register (see "Receive BD Jumbo Producer Ring Index Register (Offset 0x5870)" on page 492) may be cleared. It is unnecessary to clear both the high and low-priority mailbox registers.
- Software should write the value 0x00000000 to the low 32 bits of the Receive BD Mini Producer Ring Index mailbox (see "Receive BD Mini Producer Ring Index Register (Offset 0x278)" on page 373 for host standard and flat modes). If host software is using register indirect mode, the RX BD mini Producer ring index register (see "Receive BD Mini Producer Ring Index Register (Offset 0x5878, BCM5700 and BCM5701 Only)" on page 492) may be cleared. It is unnecessary to clear both the high and low-priority mailbox registers.

Note: Only the BCM5700 MAC with external SSRAM can use mini rings.

52. Configure the MAC unicast address. See "MAC Address Setup/Configuration" on page 167 for a full description of unicast MAC address initialization.

53. Configure random backoff seed for transmit. See the Ethernet Transmit Random Backoff register (see "Ethernet Transmit Random Backoff Register (Offset 0x438)" on page 386). Broadcom recommends using the following algorithm:

```
Seed = (MAC_ADDR[0] + MAC_ADDR[1] + MAC_ADDR[2] + MAC_ADDR[3] + MAC_ADDR[4] +
MAC_ADDR[5]) & 0x3FF
```

54. Configure the Message Transfer Unit MTU size. The MTU sets the upper boundary on RX packet size; packets larger than the MTU are marked oversized and discarded by the RX MAC. The MTU bit field in the Receive MTU Size register (see "Receive MTU Size Register (Offset 0x43C)" on page 387) must be configured before RX traffic is accepted. Host software should account for the following variables when calculating the MTU:
    - VLAN TAG
    - CRC
    - Jumbo Frames Enabled

55. Configure IPG for transmit. The Transmit MAC Lengths register (see "Transmit MAC Lengths Register (Offset 0x464)" on page 391) contains three bit fields: IPG_CRS_Length, IPG_Length, and Slot_Time_Length. The value the 0x2620 should be written into this register.

Note: An incorrectly configured IPG will introduce far end receive errors on the MAC's link partner.

56. Configure default RX return ring for non-matched packets. The MAC has a rules checker, and packets do not always have a positive match. For this situation, host software must specify a default location, where RX packet should be placed. The BCM5700/5701/5702/5703C/5703S/5704C/5704S MACs support 16 RX return rings, and software must specify a value between one and sixteen. The bit field is located in the Receive Rules Configuration register (see "Receive Rules Configuration Register (Offset 0x500)" on page 394).

57 Configure the number of Receive Lists. The Receive List Placement Configuration register (see "Receive List Placement Configuration Register (Offset 0x2010)" on page 431) allows host software to initialize QOS rules checking. For example, a value of 0x181 breaks down as follows:

- One interrupt distribution list
- Sixteen active lists
- One bad frames class

58. Write the Receive List Placement Statistics mask. Write 0xFFFFFF (24 bits) to the Receive List Placement Stats Enable Mask register (see "Receive List Placement Statistics Enable Mask Register (Offset 0x2018)" on page 433).

59. Enable RX statistics. Assert the Statistics_Enable bit in the Receive List Placement Control register (see "Receive List Placement Statistics Control Register (Offset 0x2014)" on page 432).

60. Enable the Send Data Initiator mask. Write 0xFFFFFF (24 bits) to the Send Data Initiator Enable Mask register (see "Send Data Initiator Statistics Enable Mask Register (Offset 0x0C0C)" on page 414).

61. Enable TX statistics. Assert the Statistics_Enable and Faster_Statistics_Update bits in the Send Data Initiator Control register (0x0C08)

62. Disable the host coalescing engine. Software needs to disable the host coalescing engine before configuring its parameters. Write 0x0000 to the Host Coalescing Mode register (see "Host Coalescing Mode Register (Offset 0x3C00)" on page 452).

63. Poll 20 ms for the host coalescing engine to stop. Read the Host Coalescing Mode register (see "Host Coalescing Mode Register (Offset 0x3C00)" on page 452) and poll for 0x0000. The engine was stopped in the previous step.

64. Configure the host coalescing tick count. The Receive Coalescing Ticks and Send Coalescing Ticks registers (see "Receive Coalescing Ticks Registers (Offset 0x3C08)" on page 453 and "Send Coalescing Ticks Register (Offset 0x3C0C)" on page 453) specify the number of clock ticks elapsed before an interrupt is driven. The clock begins ticking after RX/TX activity. Broadcom recommends the settings shown in Table 74.

Table 74: Recommended BCM57XX Host Coalescing Tick Counter Settings

Register	*Recommended Value*
Receive Coalescing Ticks	150
Send Coalescing Ticks	150

65. Configure the host coalescing BD count. The Receive Max Coalesced BD and Send Max Coalesced BD registers (see "Receive Max Coalesced BD Count (Offset 0x3C10)" on page 454 and "Send Max Coalesced BD Count (Offset 0x3C14)" on page 454) specify the number of frames processed before an interrupt is driven. Broadcom recommends the settings shown in Table 75.

Table 75: Recommended BCM57XX Host Coalescing Frame Counter Settings

Register	*Recommended Value*
Receive Max Coalesced Frames	10
Send Max Coalesced Frames	10

66. Configure during interrupt tick counter (BCM5700/5701/5702/5703C/5703S/5704C/5704S only). While host software processes interrupts, this value is used. See the Receive Coalescing Ticks During Interrupt and Send Coalescing Ticks During Interrupt registers (see "Receive Coalescing Ticks During Interrupt Register (Offset 0x3C18)" on page 455 and "Send Coalescing Ticks During Interrupt Register (Offset 0x3C1C)" on page 455) for further details. Broadcom recommends the settings shown in Table 76.

Table 76: Recommended BCM57XX Interrupt Tick Counter Settings

Register	**Recommended Value**
Receive Coalescing Ticks During Interrupt	0
Send Coalescing Ticks During Interrupt	0

67 Configure the max-coalesced frames during interrupt counter. While host software processes interrupts, this value is used. See the Receive Max Coalesced Frames During Interrupt and Send Max Coalesced Frames During Interrupt registers (see "Receive Max Coalesced BD Count During Interrupt (Offset 0x3C20)" on page 456 and "Send Max Coalesced BD Count During Interrupt (Offset 0x3C24)" on page 456). Broadcom recommends the settings shown in Table 77.

Table 77: Recommended BCM57XX Max Coalesced Frames During Interrupt Counter Settings

Register	**Recommended Value**
Receive Max Coalesced Frames During Interrupt	0
Send Max Coalesced Frames During Interrupt	0

68. Initialize host status block address. Host software must write a physical address to the Status Block Host Address register (see "Status Block Host Address Register (Offset 0x3C38)" on page 457), which is the location where the MAC must DMA status data. This register accepts a 64-bit value.

69. Initialize host statistics block address (BCM5700/5701/5702/5703C/5703S/5704C/5704S only). Host software must write a physical address to the Statistics Host Address register (see "Statistics Host Address Register (Offset 0x3C30)" on page 456), which is the location where the MAC must DMA statistics data. This register accepts a 64-bit value.

70. Set the statistics coalescing tick counter, which is the number of clock ticks before the MAC must DMA statistics to host physical memory (BCM5700/5701/5702/5703C/5703S/5704C/5704S only). See the Statistics Tick counter register (see "Statistics Ticks Counter Register (Offset 0x3C28)" on page 456). Broadcom recommends the setting shown in Table 78.

Table 78: Recommended BCM57XX Statistics Tick Setting

Register	**Recommended Value**
Statistics Tick	1000000

71. Configure the statistic block address in NIC local memory (BCM5700/5701/5702/5703C/5703S/5704C/5704S only). Host software should write 0x300 to the Statistics Base Address register (see "Statistics Base Address Register (Offset 0x3C40)" on page 457).

72. Configure the status block address in NIC local memory (BCM5700/5701/5702/5703C/5703S/5704C/5704S only). Host software should write 0xB00 to the Status Block Base Address register (see "Status Block Base Address Register (Offset 0x3C44)" on page 457).

73. Enable the host coalescing engine. Set the Enable bit in the Host Coalescing Mode register.

74. Enable the receive BD completion functional block. Set the Enable and Attn_Enable bits in the Receive BD Completion Mode register (see "Receive BD Completion Mode Register (Offset 0x3000)" on page 444).

75. Enable the receive list placement functional block. Set the Enable bit in the Receive List Placement Mode register (see "Receive List Placement Mode Register (Offset 0x2000)" on page 429).

76. Enable the receive list selector functional block (BCM5700/5701/5702/5703C/5703S/5704C/5704S only). Set the Enable and Attn_Enable bits in the Receive List Selector Mode register (see "Receive List Selector Mode Register (Offset 0x3400)" on page 446).

77. Enable DMA engines. Set the Enable_FHDE, Enable_RDE, and Enable_TDE bits in the Ethernet Mac Mode register (see "Ethernet MAC Mode Register (Offset 0x400)" on page 377).

78. Enable and clear statistics. Set the Clear_TX_Statistics, Enable_TX_Statistics, Clear_RX_Statistics, and Enable_TX_Statistics bits in the Ethernet Mac Mode register (see "Ethernet MAC Mode Register (Offset 0x400)" on page 377).

79 Configure the General Miscellaneous Local Control register (see "Miscellaneous Local Control Register (Offset 0x6808)" on page 507). Set the Interrupt_On_Attention bit in order for MAC to assert an interrupt whenever any of the attention bits in the CPU event register are asserted. Also set the Auto SEEPROM Access bit for MAC to access the SEEPROM through the SEEPROM address and data registers.

The configuration of GPIOs is optional and is design specific. The BCM5700 evaluation board uses the GPIO1 as a PHY reset signal. So the GPIO1 is enabled as output pin and the output on GPIO1 is driven high by asserting the following bits of 0x6808.

- Misc Pin[1] Output Enable (bit 12) — Enable GPIO1 as an output signal.
- Misc Pins[1] Output (bit 15) — Drive a logic true on GPIO_1 for PHY reset logic.

80. Write a value of zero to the Interrupt Mailbox 0 low word (see "Interrupt Mailbox 0 Register (Offset 0x200)" on page 372 for the host standard and flat modes and "Interrupt Mailbox 0 Register (Offset 0x5800)" on page 492 for the indirect mode).

81. Enable DMA completion functional block (BCM5700/5701/5702/5703C/5703S/5704C/5704S only). Set the Enable bit in the DMA Completion Mode register (see "DMA Completion Mode Register (Offset 0x6400)" on page 500).

82. Configure the Write DMA Mode register (see "Write DMA Mode Register (Offset 0x4C00)" on page 480). The following bits are asserted:

- Enable—starts the functional block
- Write_DMA_PCI_Target_Abort_Attention_Enable
- Write_DMA_PCI_Master_Abort_Attention_Enable
- Write_DMA_PCI_Parity_Attention_Enable
- Write_DMA_PCI_Host_Address_Overflow_Attention_Enable
- Write_DMA_PCI_FIFO_Underrun_Attention_Enable
- Write_DMA_PCI_FIFO_Overrun_Attention_Enable
- Write_DMA_PCI_FIFO_Overread_Attention_Enable
- Write_DMA_Local_Memory_Read_Longer_Than_DMA_Length

83. Configure the Read DMA Mode register (see "Read DMA Mode Register (Offset 0x4800)" on page 477). The following bits are asserted:

- Enable—start functional block
- Read_DMA_PCI_Target_Abort
- Read_DMA_PCI_Master_Abort
- Read_DMA_PCI_Parity_Error
- Read_DMA_PCI_Host_Overflow_Error
- Read_DMA_PCI_FIFO_Overrun_Error
- Read_DMA_PCI_FIFO_Underrun_Error
- Read_DMA_PCI_FIFO_Overread_Error
- Read_DMA_Local_Memory_Write_Longer_Than_DMA_Length

84. Enable the receive data completion functional block. Set the Enable and Attn_Enable bits in the Receive Data Completion Mode register (see "Receive Data Completion Mode Register (Offset 0x2800)" on page 441).

85. Enable the Mbuf cluster free functional block (BCM5700/5701/5702/5703C/5703S/5704C/5704S only). Set the Enable bit in the Mbuf Cluster Free Mode register (see "MBUF Cluster Free Mode Register (Offset 0x3800)" on page 447).

86. Enable the send data completion functional block. Set the Enable bit in the Send Data Completion Mode register (see "Send Data Completion Mode Register (Offset 0x1000)" on page 420).

87. Enable the send BD completion functional block. Set the Enable and Attn_Enable bits in the Send BD Completion Mode register (see "Send BD Completion Mode Register (Offset 0x1C00)" on page 425).

88. Enable the Receive BD Initiator Functional Block. Set the Enable and Receive_BDs_Available_On_Receive_BD_Ring in the Receive BD Initiator Mode register (see "Receive BD Initiator Mode Register (Offset 0x2C00)" on page 442).

89. Enable the receive data and BD initiator functional block. Set the Enable and Illegal_Return_Ring_Size bits in the Receive Data and Receive BD Initiator Mode register (see "Receive Data and Receive BD Initiator Mode Register (Offset 0x2400)" on page 437).

90. Enable the send data initiator functional block. Set the Enable bit in the Send Data Initiator Mode register (see "Send Data Initiator Mode Register (Offset 0x0C00)" on page 412).

91. Enable the send BD initiator functional block. Set the Enable and Attn_Enable bits in the Send BD Initiator Mode register (see "Send BD Initiator Mode Register (Offset 0x1800)" on page 424).

92. Enable the send BD selector functional block. Set the Enable and Attn_Enable bits in the Send BD Selector Mode register (see "Send BD Ring Selector Mode Register (Offset 0x1400)" on page 422).

93. Download firmware (optional). See "Firmware Download" on page 162.

94. Enable the transmit MAC. Set the Enable bit in the Transmit MAC Mode register (see "Transmit MAC Mode Register (Offset 0x45C)" on page 390). Optionally, software may set the Enable_Flow_Control to enable 802.3x flow control.

95. Enable the receive MAC. Set the Enable bit in the Receive MAC Mode register (see "Receive MAC Mode Register (Offset 0x468)" on page 391). Optionally, software may set the following bits:
    - Enable_Flow_Control—enable 802.3x flow control
    - Accept_oversized—ignore RX MTU up to 64K maximum size
    - Promiscuous_Mode—accept all packets regardless of dest address
    - No_CRC_Check—RX MAC will not check Ethernet CRC

96. Disable auto-polling on the management interface (optional) by writing 0xC0000 to the MI Mode register (see "MI Mode Register (Offset 0x454)" on page 389).

Note: Broadcom recommends using PHY interrupts for link status change indications. Auto-polling is another mechanism for determining link status change (see "PHY Setup and Initialization" on page 250).

97. Configure D0 power state in PMSCR. See "Power Management Control/Status Register (Offset 0x4C)" on page 318. Optional—the PMCSR register is reset to 0x00 after chip reset. Software may optionally reconfigure this register if the device is being moved from D3 hot/cold.

98. Program Hardware to control LEDs. Write 0x00 to LED Controls register. LEDs on the BCM57XX reference designs are tied to the physical layer.

99. Activate link and enable MAC functional blocks. Set the Link_Status bit in the MI Status register (see "MI Status Register (Offset 0x450)" on page 389) to generate a link attention.

100. Setup the physical layer and restart auto-negotiation. For details on PHY auto-negotiation, refer to the PHY data sheet. (The PHY core used in each MAC is listed in Table 2 on page 5.). See "PHY Setup and Initialization" on page 250 for information on setting up and initializing the PHY.

101. Setup multicast filters. Refer to "Packet Filtering" on page 168 for details on multicast filter setup.

102. Enable interrupts. Clear the Mask_PCI_Interrupt_Output bit in the Miscellaneous Host Control register (see "Miscellaneous Host Control Register (Offset 0x68)" on page 325).

SHUTDOWN

To power down the BCM57XX family, its state machines must be disabled in specific sequence as shown below. The host software must clear the Enable bit of each state machine and poll the bit until it is cleared. The maximum poll period that software must wait for the enable bits to clear is 2 ms; except, the read and write DMA mode registers require a maximum timeout of 4 ms.

```
// Receive path shutdown sequence.
Receive_MAC_Mode.Enable = 0                                     // 0x0468
Receive_BD_Initiator_Mode.Enable = 0                            // 0x2c00
Receive_List_Placement_Mode.Enable = 0                          // 0x2000
Receive_List_Selector_Mode.Enable = 0
(BCM5700/5701/5702/5703C/5703S/5704C/5704S only)                // 0x3400
Receive_Data_BD_Initiator_Mode.Enable = 0                       // 0x2400
Receive_Data_Completion_Mode.Enable = 0                         // 0x2800
Receive_BD_Completion_Mode.Enable = 0                           // 0x3000

// Transmit path shutdown sequence.
Send_BD_Selector_Mode.Enable = 0                                // 0x1400
Send_BD_Initiator_Mode.Enable = 0                               // 0x1800
Send_Data_Initiator_Mode.Enable = 0                             // 0x0c00
Read_DMA_Mode.Enable = 0                                        // 0x4800
Send_Data_Completion_Mode.Enable = 0                            // 0x1000
DMA_Completion_Mode.Enable = 0
(BCM5700/5701/5702/5703C/5703S/5704C/5704S only)                // 0x6400
Send_BD_Completion_Mode.Enable = 0                              // 0x1c00
MAC_Mode Register TDE bit (bit 21)= 0                           // 0x0400
Transmit_MAC_Mode.Enable = 0                                    // 0x045C

// Memory related state machines shutdown.
Host_Coalescing_Mode.Enable = 0                                 // 0x3c00
DMA_Write_Mode.Enable = 0                                       // 0x4c00
MBUF_Cluster_Free_Mode.Enable = 0
(BCM5700/5701/5702/5703C/5703S/5704C/5704S only)                // 0x3800
FTQ_Reset = 0xffffffff                                          // 0x5c00
FTQ_Reset = 0                                                   // 0x5c00
Buffer_Manager_Mode.Enable = 0
(BCM5700/5701/5702/5703C/5703S/5704C/5704S only)                // 0x4400
Memory_Arbiter_Mode.Enable = 0
(BCM5700/5701/5702/5703C/5703S/5704C/5704S only)                // 0x4000
```

Note: The Buffer Manager and Memory Arbiter should not be disabled as part of shutdown of BCM5705/5788/5721/5751/5714C/5714S/5715C/5715S/5752 MACs. This is because the scratch pad memory for the on-chip RISC processor of these MACs is reserved out of the RxMbuf memory space.

RESET

A hardware reset can be initiated by the PCI reset signal or by the host software via the Core Clock Blocks Reset bit. Such a reset will initialize all PCI configuration registers to their default values, though the boot code may then modify some values such as those listed below. The content of the device internal memory remains unchanged after reset.

At the end of the reset, the RX RISC executes a small on chip ROM code. This code loads an executable image contained in an attached NVRAM and referred to as the bootcode. This bootcode allows at least the following fields to be initialized to different values to support product variations (for additional details, see "NVRAM Configuration" on page 88).

- Vendor ID
- Device ID
- Subsystem Vendor ID
- Subsystem Device ID
- Possible PHY initialization

The bootcode may have additional functionalities such as PXE that must be acquiesced while the host software is running. For instance, an NDIS driver issues a hardware reset via the Core Clock Blocks Reset bit. After the reset is completed, the RX RISC begins executing the bootcode as if the power was first applied to the device. However, the Ndis driver must have a mechanism to prevent the PXE driver from running and the bootcode must be able to distinguish between a power-on reset and a reset initiated by the host software. The host software and the bootcode could implement a reset handshake by using shared memory at offset 0x0b50 as a software mailbox (see "Firmware Mailbox" on page 275).

FIRMWARE DOWNLOAD

FIRMWARE BINARY IMAGE

The RISC cores in the BCM57XX family of chips execute the MIPS-2 instruction set. Broadcom uses a GNU tool kit to create the firmware binary code. The output from the GNU build is a C language header file, which contains the machine code for the embedded RISC cores. This manual does not cover the technology necessary to program the RISC cores. However, programmers may need to download value-added firmware provided by Broadcom. One example of value added firmware is TCP segmentation firmware that can be loaded from the host driver. This section provides the necessary understanding of the header file, created by the Broadcom GNU tools. The programmer must understand the layout of the header file, to accomplish a firmware download.

The following sections are located in the header file provided by Broadcom:

- t3FwText[]—Array of 32-bit words. This section contains the machine code (opcodes/operands) executed by the RISC cores. This is the text section in the binary file, output by the GNU build process.
- t3FwRodata[]—Array of 32-bit words. This section contains the read-only data available to the code section of the firmware.
- t3FwData[]—Array of 32-bit words. This section contains the local variables available to the code section of the firmware.

The programmer needs to move these sections to both the RX/TX scratchpads for the BCM5700 through BCM5704, and to the scratchpad that is reserved out of RXMBUF for the BCM5705, BCM5788, BCM5721, BCM5751, BCM5714C, BCM5714S, BCM5715C, BCM5715S, and BCM5752. This is the binary image that the RISC core executes. The following figure shows a conceptual layout of the scratchpad/RXMBUF and how the header file sections are moved to the scratchpads/RXMBUF. Three sections are moved.

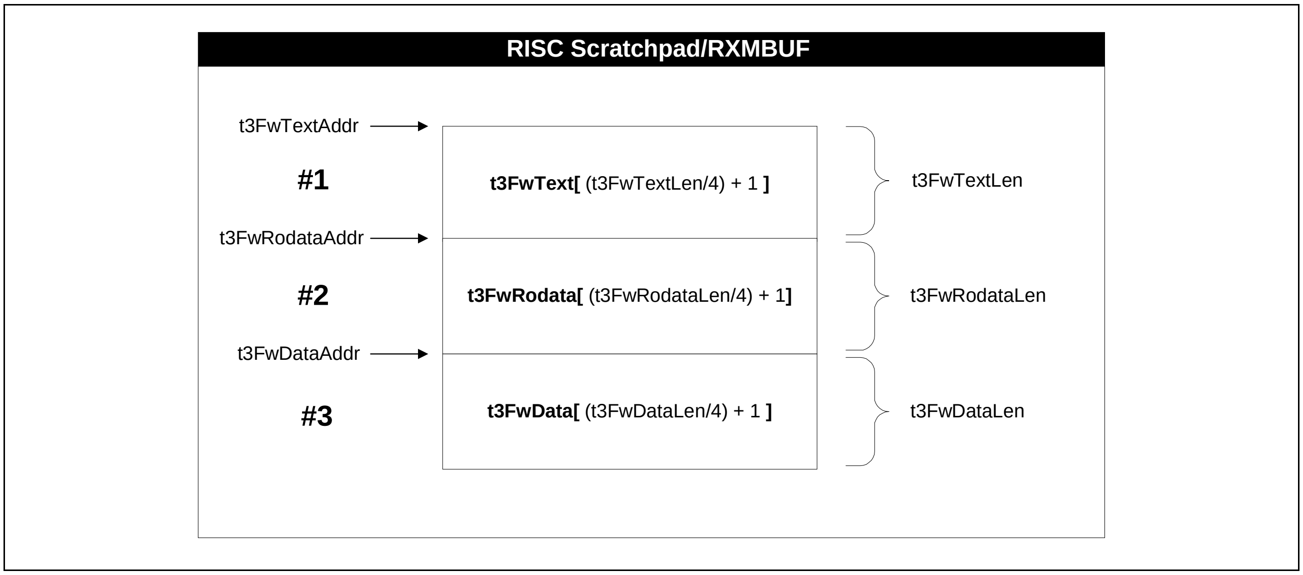


Figure 62: Firmware Image Moved to Scratchpad/RXMBUF

The programmer should be careful when moving the header file data to the scratchpad/RXMBUF. The variables t3FwDataAddr, t3FwRodataAddr, and t2FwTextAddr are all RISC Core relative addresses; the host must translate each address to a BCM57XX family's register space region. The host programmer must use the address offset from 0x30000 (RX RISC), the offset from 0x34000 (TX RISC), or the offset from 0x10000 (RXMBUF). In summary, the address offsets in the header file are relative to the local memory address ranges specified in Table 79 for the RX and TX RISC respectively. The following formula may be used to convert address data:

```
RX Register Address = 0x30000 + (fwHeaderFileAddress & 0xFFFF)
TX Register Address = 0x34000 + (fwHeaderFileAddress & 0xFFFF)
RXMBUF Address = 0x10000 + (fwHeaderFileAddress & 0xFFFF)
```

Table 79: Addressing Perspectives

Memory Type	***Host Perspective***	***RX RISC Perspective***	***TX RISC Perspective***
RX RISC scratchpad	0x30000–0x33FFF	0x08000000–0x08003FFF	0xC0030000–0xC0033FFF
TX RISC scratchpad	0x34000–0x37FFF	0xC0034000–0xC0037FFF	0x080000000–0x08003FFF
RXMBUF	0x10000–0x1DFFF	0x10000–0x1DFFF	N/A(a) (a) = Devices that use RXMBUF memory do not have TX RISC processors.
Internal Memory	0x000000–0x01FFFF	0x00000000–0x0001FFFF	0x00000000–0x0001FFFF
External Memory	0x020000–0xFFFFFF	0x00020000–0x00FFFFFF	0x0020000–0x00FFFFFF

RESET RISC PROCESSOR

The RX processor can be reset by setting the Reset RX RISC Bit of the RX RISC Mode register (see "RX RISC Mode Register (Offset 0x5000)" on page 483). Similarly, the TX processor can be reset by setting the Reset TX RISC Bit of the TX RISC Mode register (see "TX RISC Mode Register (Offset 0x5400)" on page 487). This bit is self-clearing bit; it will be cleared once internal reset of processor is completed.

For example, to reset RX RISC, do the following:

```
WR 5004, 0xffffffff /* Clear all CPU state */
WR 0x5000, 0x1
Wait until Bit 0 of register at 0x5000 is cleared.
```

To reset the TX RISC, do the following:

```
WR 5404, 0xffffffff /* Clear all CPU state */
WR 0x5400, 0x1
Wait until Bit 0 of register at 0x5000 is cleared.
```

Halt RISC Procedure

1. Clear the RX/TX RISC state register. Write 0xFFFFFFFF to the RX_RISC_State register (see "RX RISC State Register (Offset 0x5004)" on page 485) or the TX_RISC_State register (see "TX RISC State Register (Offset 0x5404)" on page 488) respectively.

2. Issue RX/TX RISC halt. Write the RISC_MODE_HALT bit to the RX_RISC_Mode register (see "RX RISC Mode Register (Offset 0x5000)" on page 483) or the TX_RISC_Mode register (see "TX RISC Mode Register (Offset 0x5400)" on page 487) respectively.

3. Read/verify that the RISC_MODE_HALT bit is set. Read the Rx/Tx RISC_MODE_HALT bit back from the RX_RISC_Mode or TX_RISC_Mode register respectively. Break from procedure if bit is set.

4. Delay 10 µs and jump to step 1. Repeat the procedure up to 10,000 times.

Start RISC Procedure

This procedure stops the RX/TX RISC CPUs and modifies the program counter to begin executing firmware at a new address.

1. Clear the RX/TX RISC state register. Write 0xFFFFFFFF to the RX_RISC_State register (see "RX RISC State Register (Offset 0x5004)" on page 485) or the TX_RISC_State register (see "TX RISC State Register (Offset 0x5404)" on page 488) respectively.

2. Set the RX/TX RISC program counter. Write t3FwTextAddr to the RX_RISC_PC register (see "RX RISC Program Counter (Offset 0x501C)" on page 486) or the TX_RISC_PC register (see "TX RISC Program Counter (Offset 0x541C)" on page 489) respectively.

Note: The t3FwTextAddr should not be converted to a register relative address. The RISCs execute from a local memory space. The conversion is only necessary for writing t3FwText [] to the scratchpad using register space—the host view of the scratchpad region is different from the RISC view. See Table 79 on page 163.

3. Read back the PC register. Read the RX_RISC_PC or TX_RISC_PC register respectively and verify that t3FwTextAddr is set. If properly set, then jump to step 7.

4. Clear the RX/TX RISC state register. Write 0xFFFFFFFF to the RX_RISC_State or TX_RISC_State register respectively.

5. Halt the RX/TX RISC. Write the RISC_MODE_HALT bit to the RX_RISC_Mode or TX_RISC_Mode register respectively.

6 Delay one millisecond. Jump to step 2 and repeat procedure.

7. Clear the RX/TX RISC state register. Write 0xFFFFFFFF to the RX_RISC_State or TX_RISC_State register respectively.

8. Clear the RX/TX RISC mode register. Write 0x00 to the RX_RISC_Mode register (see "RX RISC Mode Register (Offset 0x5000)" on page 483) or the TX_RISC_Mode register (see "TX RISC Mode Register (Offset 0x5400)" on page 487) respectively.

Firmware Download Procedure

The host driver should use register indirect access to modify both the scratchpad and RISC register space. See "Pseudocode" on page 205 in Section 9: "PCI".

1. Halt the RX RSIC Core (see "Halt RISC Procedure" on page 164).
2. Optional for BCM5700, BCM5701, BCM5702, BCM5703C, BCM5703S, BCM5704C, and BCM5704S—clear the RX RISC Scratchpad. Use register indirect access and write zero(s) starting at register address 0x30000 (see Table 79 on page 163). The last address to clear is 0x33FFF. The total length of the scratchpad is 0x4000 and the host driver should increment the target address by four (4), since each MIPS word is 32 bits (4 bytes).
3. Convert variable t3FwRodataAddr to a register relative address for the RX RISC. The register address is calculated as follows:
   - For BCM5705, BCM5788, BCM5721, BCM5751, BCM5714C, BCM5714S, BCM5715C, BCM5715S, and BCM5752:
     ```
     regNormalized = 0x10000 + (t3FwTextAddr & 0xFFFF)
     ```
   - For BCM5700, BCM5701, BCM5702, BCM5703C, 5703S, BCM5704C, and BCM5704S:
     ```
     regNormalized = 0x30000 + (t3FwTextAddr & 0xFFFF)
     ```
   - For systems using RXMBUF memory for firmware, ensure that the downloaded firmware does not overlap the MBUF Pool Address.
4. Write the array t3FwText [] to the RX scratchpad/RXMBUF (see Table 79 on page 163). Use register indirect access and increment by four bytes for every 32-bit write. The last/limit address to write is calculated as follows:

```
regNormalized (Step 3.) + t3FwTextLen
```

5. Convert variable t3FwRodataAddr to a register relative address for the RX RISC. The register address is calculated as follows:
   - For BCM5705, BCM5788, BCM5721, BCM5751, BCM5714C, BCM5714S, BCM5715C, BCM5715S, and BCM5752:
     ```
     regNormalized = 0x10000 + (t3FwRoDataAddr & 0xFFFF)
     ```
   - For BCM5700, BCM5701, BCM5702, BCM5703C, 5703S, BCM5704C, and BCM5704S:
     ```
     regNormalized = 0x30000 + (t3FwRoDataAddr & 0xFFFF)
     ```
6. Write the array t3FwRodata [] to the RX scratchpad/RXMBUF (see Table 79 on page 163). Use register indirect access and increment by four bytes for every 32-bit write. The last/limit address to write is calculated as follows:

```
regNormalized (from Step 5.) + t3FwRodataLen
```

7. Convert variable t3FwDataAddr to a register relative address for the RX RISC. The register address is calculated as follows:
   - For BCM5705, BCM5788, BCM5721, BCM5751, BCM5714C, BCM5714S, BCM5715C, BCM5715S, and BCM5752:
     ```
     regNormalized = 0x10000 + (t3FwDataAddr & 0xFFFF)
     ```
   - For BCM5700, BCM5701, BCM5702, BCM5703C, 5703S, BCM5704C, and BCM5704S:
     ```
     regNormalized = 0x30000 + (t3FwDataAddr & 0xFFFF)
     ```
8. Write the array t3FwData [] to the RX scratchpad/RXMBUF (see Table 79 on page 163). Use register indirect access and increment by four bytes for every 32-bit write. The last/limit address to write is calculated as follows:

```
regNormalized (from Step 7.) + t3FwDataLen
```

Note: Steps 9 through 18 are only required for firmware that runs on both the TX and RX CPUs. The BCM5705, BCM5788, BCM5721, BCM5751, BCM5714C, BCM5714S, BCM5715C, BCM5715S, and BCM5752 MACs have only RX RISC and hence the steps 9 through 18 are not applicable.

9. Reset the TX RISC Core (see Table 79 on page 163).
10. Clear the TX RISC Scratchpad. Use register indirect access and write zero(s) starting at register address 0x34000 (see Table 79 on page 163). The last address to clear is 0x37FFF. The total length of the scratchpad is 0x4000 and the host driver should increment the target address by four (4), since each MIPS word is 32 bits (4 bytes).
11. Clear the TX RISC state register. Use register indirect access and write 0xFFFFFFFF to the TX_RISC_State_Register (see "TX RISC State Register (Offset 0x5404)" on page 488).
12. Halt the TX RISC. Assert the Halt_TX_CPU bit in the TX_RISC_Mode register (see "TX RISC Mode Register (Offset 0x5400)" on page 487).
13. Convert variable t3FwRodataAddr to a register relative address for the TX RISC. The register address is calculated as follows:

```
regNormalized = 0x34000 + (t3FwTextAddr & 0xFFFF)
```

14. Write the array t3FwText [] to the TX scratchpad (see Table 79 on page 163). Use register indirect access and increment by four bytes for every 32-bit write. The last/limit address to write is calculated as follows:

```
regNormalized (from Step 13.) + t3FwTextLen
```

15. Convert variable t3FwRodataAddr to a register relative address for the TX RISC. The register address is calculated as follows:

```
regNormalized = 0x34000 + (t3FwRodataAddr & 0xFFFF)
```

16. Write the array t3FwRodata [] to the TX scratchpad (see Table 79 on page 163). Use register indirect access and increment by four bytes for every 32-bit write. The last/limit address to write is calculated as follows:

```
regNormalized (from Step 15.) + t3FwRodataLen
```

17. Convert variable t3FwDataAddr to a register relative address for the TX RISC. The register address is calculated as follows:

```
regNormalized = 0x34000 + (t3FwDataAddr & 0xFFFF)
```

18. Write the array t3FwData [] to the TX scratchpad (see Table 79 on page 163). Use register indirect access, and increment by four bytes for every 32-bit write. The last/limit address to write is calculated as follows:

```
regNormalized (from Step 17.) + t3FwDataLen
```

19. Start the RX RISC (see "Start RISC Procedure" on page 164).

Note: When enabled, the ASF/IPMI FW needs to run even in the absence of OS. The IPMI/ASF FW is normally programmed into NVRAM and the boot code will load this FW into scratchpad/RxMbuf memory if the IPMI/ASF feature is enabled. The size of IPMI/ASF FW is larger than 16 KB (size of scratchpad memory area in BCM5700 through BCM5704 devices) and therefore it is run on both the Tx and Rx RISCs for the BCM5700 through BCM5704 devices. It is only necessary to start the Rx RISC because the IPMI/ASF firmware code executed by the Rx RISC starts the Tx RISC. In case of BCM5705/5788/5721/5751/5714C/5714S/5715C/5715S/5752 devices that have only one RISC per port, the bootcode reserves enough memory out of RxMbuf as RISC scratchpad for IPMI/ASF firmware.

The TCP Segmentation (aka LargeSendOffload) FW can be loaded from the driver, but it cannot coexist with the IPMI/ASF FW on a given port even in BCM5700 through BCM5704 devices that have two RISCs per port because the IPMI/ASF is run on both RISCs of these devices. In case of dual-port MACs like BCM5704, it should be possible to run LSO on one port and IPMI/ASF on another port.

When freeing received RxMbufs (e.g., received ASF RMCP packets) from the MIPS CPU (running ASF firmware or any other optional FW) while simultaneously receiving/freeing TCP traffic for the host device driver, a race condition can occur which causes subsequent receives to fail (no more RxMbufs get enqueued). To avoid this race condition, before freeing the RxMbuf chain (by writing to the MbufClustFreeFtqFifoEnqueueDequeue register, 0x5cc8), the FW should poll the buffer manager hardware diagnostic 3 register (0x4454) until the descriptor in bits 25-16 matches the MBUF cluster at the beginning of the chain we're about to free. Also, note that the hardware diagnostic register uses a cluster based on the offset from the top of the RxMbuf pool (e.g., 0x16000) and the firmware uses a cluster based on the offset from 0x10000, so some calculations must be performed to compare the proper pointer values.

Example Code Snippet (from ASF Firmware)

```
/* Poll buffer manager hardware diag register 3 (0x4454) for our mbuf pointer value */
while(trp->BufMgr.MbufPoolAddr + (((trp->BufMgr.Hwdiag[2]>>16) & 0x1ff) << 7) !=
(u32)pmbuf);
```

MAC ADDRESS SETUP/CONFIGURATION

The MAC address registers, starting at offset 0x0410, contain the MAC addresses of the NIC. These registers are usually initialized with a default MAC address extracted from the NIC's NVRAM when it is first powered up. The host software may overwrite the default MAC address by writing to the MAC registers with a new MAC address. Table 80 illustrates the MAC register format.

The BCM57XX family allows a NIC to have up to four MAC addresses (offset 0x410-0x42F) that are used for hardware packet reception filtering. However, most host software will initialize the registers of the four MAC addresses to the same MAC address since a NIC usually has only one MAC address. An additional 12 MAC addresses are supported for the BCM5702, BCM5703C, BCM5703S, BCM5704C, and BCM5704S MACs. They are accessed at registers 0x530 to 0x58f, and have the same format as registers 0x410 to 0x42f shown in Table 80 below.

The host software could set up the send buffer descriptors of a frame to overwrite the source address with one of the four MAC addresses. "Send Buffer Descriptors" on page 94 for more details.

When flow control is enabled on the BCM57XX family, the MAC Address 0 is used as the source address for sending PAUSE frames (see "Pause Control Frame" on page 707).

Table 80: Mac Address Registers

Register Name	*Offset*	*31*	*24*	*23*	*16*	*15*	*8*	*7*	*0*
Mac_Address_0	0x0410	Unused				Octet 0		Octet 1	
	0x0414	Octet 2		Octet 3		Octet 4		Octet 5	
Mac_Address_1	0x0418	Unused				Octet 0		Octet 1	
	0x041c	Octet 2		Octet 3		Octet 4		Octet 5	
Mac_Address_2	0x0420	Unused				Octet 0		Octet 1	
	0x0424	Octet 2		Octet 3		Octet 4		Octet 5	
Mac_Address_3	0x0428	Unused				Octet 0		Octet 1	
	0x042c	Octet 2		Octet 3		Octet 4		Octet 5	

PACKET FILTERING

MULTICAST HASH TABLE SETUP/CONFIGURATION

The MAC hash registers are used to help discard unwanted multicast packets as they are received from the external media. The destination address is fed into the normal CRC algorithm in order to generate a hash function. The most significant bits of the CRC are then used without any inversion in reverse order to index into a hash table, which is comprised of these MAC hash registers. If the CRC is calculated by shifting right, then the right-most bits of the CRC can be directly used with no additional inversion or bit swapping required. See "Ethernet CRC Calculation" for more details on the CRC algorithm.

All four MAC hash registers are used so that register 1 bit-32 is the most significant hash table entry and register 4 bit-0 is the least significant hash table entry. This follows the normal big-endian ordering used throughout the BCM57XX family. Since there are 128 hash table entries, 7 bits are used from the CRC.

The MAC hash registers are ignored if the receive MAC is in promiscuous mode.

ETHERNET CRC CALCULATION

The BCM57XX family uses the standard 32-bit CRC required by the Ethernet specification as its FCS in all packets. The checksum is the 32-bit remainder of the polynomial division of the data taken as a bit stream of polynomial coefficients and a predefined constant, which also represents binary polynomial coefficients. The checksum is optionally appended most-significant bit first to a packet, which is to be sent down the wire. At the receiving side, the division is repeated on the entire packet including the CRC checksum. The remainder is compared to a known constant. For details on the mathematical basis for CRC checksums, see Tanenbaum's Computer Networks, Third Edition, c1996.

The 32-bit CRC polynomial divisor is shown below:

```
x32 + x26 + x23 + x22 + x16 + x12 + x11 + x10 + x8 + x7 + x5 + x4 + x2 + x + 1
```

GENERATING CRC

The following steps describe a method to calculate the CRC with the resulting 32-bit quantity having reversed bit order (i.e., most significant bit x31 of the remainder is right-most bit). The data should be treated as a stream of bytes. Set remainder to 0xFFFFFFFF. For each bit of data starting with least-significant bit of each byte:

1. If right-most bit (bit-0) of the current remainder XOR'd with the data bit equal 1,
   then remainder = (remainder shifted right one bit) XOR 0xEDB88320,
   else remainder = (remainder shifted right one bit).
2. Invert remainder such that remainder = ~remainder.
   Remainder is CRC checksum.
   Right-most byte is the most significant and is to be sent first.
   Swap bytes of CRC if big-endian byte ordering is desired.

CHECKING CRC

The following steps describe a method to check a stream of bytes, which has a CRC appended.

1. Set remainder to 0xFFFFFFFF.
2. For each bit of data starting with least-significant bit of each byte:
   If right-most bit (bit-0) of the current remainder XOR'd with the data bit equal 1,
   then remainder = (remainder shifted right one bit) XOR 0xEDB88320,
   else remainder = (remainder shifted right one bit).
3. Remainder should equal magic value 0xDEBB20E3 if CRC is correct.

INITIALIZING THE MAC HASH REGISTERS

The 128-bit multicast hash table is treated as a single object occupying four BCM57XX registers starting at offset 0x0470 (see Table 81). The 128-bit value follows the big-endian ordering required by BCM57XX family. Thus, the most significant 32-bit of the 128-bit value resides in Mac_Hash_Register_0 at offset 0x0470 and the least significant 32-bit resides in Mac_Hash_Register_3 at offset 0x047c.

Host software can enable the reception of all multicast frames including broadcast frames by setting all four multicast hash registers to 0xFFFFFFFF.

Table 81: Multicast Hash Table Registers

Register Name	*Offset*	*Description*
Mac_Hash_Register_0	0x0470	Most significant 32-bit of the 128-bit hash table
Mac_Hash_Register_1	0x0474	Bit 64-93 of the 128-bit hash table
Mac_Hash_Register_2	0x0478	Bit 32-63 of the 128-bit hash table
Mac_Hash_Register_3	0x047c	Least significant 32-bit of the 128-bit hash table

The C code fragment below will illustrate how to initialize the multicast hash table registers. The code fragment computes the indices into hash table from a given list of multicast addresses and initializes the multicast hash registers.

```
Unsigned long HashReg[4];
Unsigned long j, McEntryCnt;
Unsigned char McTable[32][6]; // List of multicast addresses to accept.

// Initialize the McTable here.
McEntryCnt = 32;

// Initialize the multicast table registers.
HashReg[0] = 0;               // Mac_Hash_Regsiter_0 at offset 0x0470.
HashReg[1] = 0;               // Mac_Hash_Register_1 at offset 0x0474.
HashReg[2] = 0;               // Mac_Hash_Register_2 at offset 0x0478.
HashReg[3] = 0;               // Mac_Hash_Register_3 at offset 0x047c.

for(j = 0; j < McEntryCnt; j++)
{
   unsigned long RegIndex;
   unsigned long Bitpos;
   unsigned long Crc32;

   Crc32 = ComputeCrc32(McTable[j], 6);

   // The most significant 7 bits of the CRC32 (no inversion),
    // are used to index into one of the possible 128 bit positions.
    Bitpos = ~Crc32 & 0x7f;

    // Hash register index.
    RegIndex = (Bitpos & 0x60) >> 5;

    // Bit to turn on within a hash register.
    Bitpos &= 0x1f;

    // Enable the multicast bit.
    HashReg[RegIndex] |= (1 << Bitpos);

}
```

The C routine below computes the Ethernet CRC32 value from a given byte stream. The routine is called from the above code fragment.

```
// Routine for generating CRC32.
unsigned long
ComputeCrc32(
    unsigned char *pBuffer, // Buffer containing the byte stream.
    unsigned long BufferSize) // Size of the buffer.
{
    unsigned long Reg;
    unsigned long Tmp;
    unsigned long j, k;

    Reg = 0xffffffff;

    for(j = 0; j < BufferSize; j++)
    {
        Reg ^= pBuffer[j];

        for(k = 0; k < 8; k++)
        {
            Tmp = Reg & 0x01;
            Reg >>= 1;
            if(Tmp)
            {
                Reg ^= 0xedb88320;
                }
            }
    }

    return ~Reg;
}
```

Promiscuous Mode Setup/Configuration

The host software may enable promiscuous mode by setting the Promiscous_Mode bit (bit 8) of the Receive_MAC_Mode register (offset 0x468). The Promiscous_Mode bit defaults to disabled after reset, and host software must explicitly set this bit for promiscuous mode. In promiscuous mode of operation, the NetXtreme BCM57XX family accepts all incoming frames that are not filtered by the active receive rules regardless of the destination MAC address. In other words, the NetXtreme BCM57XX MAC operating in promiscuous mode ignores multicast and MAC address filtering ("Multicast Hash Table Setup/Configuration" on page 168 and "MAC Address Setup/Configuration" on page 167) but applies Receive Rules.

Broadcast Setup/Configuration

The host software may configure the BCM57XX family to discard the received broadcast frames by using two receive rules as defined below. The NetXtreme BCM57XX family parses all incoming frames according to these receive rules and discards those frames that have a broadcast destination address (see "Receive Rules Setup and Frame Classification" on page 126 for more details on setting up the receive rules).

Below are the two receive rules for discarding broadcast frames.

```
Rule1 Control: 0xc2000000          Rule1 Mask/Value: 0xffffffff
Rule2 Control: 0x86000004          Rule2 Mask/Value: 0xffffffff
```

Memory Maps and Pool Configuration

The BCM57XX NetXtreme family provides 128K of internal SRAM. The first 32K of internal SRAM is called Miscellaneous Memory and it is used internally by the device for Ring buffers, Statistics, and Status block. The MAC allocates the remaining SRAM as Memory Buffers (or MBUFs) for packet buffers and firmware value-adds. (For the amount of MBUF space allocated for each member of the BCM57XX family, see Table 1 on page 2.). The BCM5700 NetXtreme MAC also supports external SSRAM option. The following configuration is possible when the BCM57XX hardware design/application provides only internal memory:

- Receive standard producer ring enabled
- Receive jumbo producer ring enabled
- Send rings 0–3 enabled

Table 82: NetXtreme Mem. Map (5700 (Int. SRAM)/5701/5702/5703C/5703S/5704C/5704S MAC only)

Region	Size	NIC RISC Processor View	Host Flat View	Host Standard View	Host UNDI View
Page zero	256B	0x00000000– 0x000000FF	0x01000000– 0x010000FF	0x00000000–[a] 0x000000FF	0x00000000–[a] 0x000000FF
Send ring RCB	256B	0x00000100– 0x000001FF	0x01000100– 0x010001FF	0x00000100–[a] 0x000001FF	0x00000100–[a] 0x000001FF
Receive return ring RCB	256B	0x00000200– 0x000002FF	0x01000200– 0x010002FF	0x00000200–[a] 0x000002FF	0x00000200–[a] 0x000002FF
Statistics block	2 KB	0x00000300– 0x00000AFF	0x01000300– 0x01000AFF	0x00000300–[a] 0x00000AFF	0x00000300–[a] 0x00000AFF
Status block	80B	0x00000B00– 0x00000B4F	0x01000B00– 0x01000B4F	0x00000B00–[a] 0x00000B4F	0x00000B00–[a] 0x00000B4F
Software Gencomm	1 KB	0x00000B50– 0x00000FFF	0x01000B50– 0x01000FFF	0x00000B50–[a] 0x00000FFF	0x00000B50–[a] 0x00000FFF
Unmapped[b]	4 KB	0x00001000– 0x00001FFF	0x01001000– 0x01001FFF	0x00001000–[a] 0x00001FFF	0x00001000–[a] 0x00001FFF
DMA descriptors	8 KB	0x00002000– 0x00003FFF	0x01002000– 0x01003FFF	0x00002000–[a] 0x00003FFF	0x00002000–[a] 0x00003FFF

Table 82: NetXtreme Mem. Map (5700 (Int. SRAM)/5701/5702/5703C/5703S/5704C/5704S MAC only) (Cont.)

Region	Size	NIC RISC Processor View	Host Flat View	Host Standard View	Host UNDI View
Send rings 1–4	8 KB	0x00004000– 0x00005FFF	0x01004000– 0x01005FFF	0x00004000–[a] 0x00005FFF	0x00004000–[a] 0x00005FFF
Standard receive rings	4 KB	0x00006000– 0x00006FFF	0x01006000– 0x01006FFF	0x00006000–[a] 0x00006FFF	0x00006000–[a] 0x00006FFF
Jumbo receive rings	4 KB	0x00007000– 0x00007FFF	0x01007000– 0x01007FFF	0x00007000–[a] 0x00007FFF	0x00007000–[a] 0x00007FFF
Buffer pool 1[c]	32 KB	0x00008000– 0x0000FFFF	0x01008000– 0x0100FFFF	0x00008000–[a] 0x0000FFFF	0x00008000–[a] 0x0000FFFF
Buffer pool 2 or (expansion ROM)[d]	32 KB	0x00010000– 0x00017FFF	0x01010000– 0x01017FFF	0x00010000–[a] 0x00017FFF	0x00010000–[a] 0x00017FFF
Buffer pool 3[e] or (expansion ROM)[d]	32 KB	0x00018000– 0x0001FFFF	0x01018000– 0x0101FFFF	0x00018000–[a] 0x0001FFFF	0x00018000–[a] 0x0001FFFF

[a] Indirect access via Memory Window Base Address and Memory Window Data registers pair.
[b] Read access to unmapped memory returns unexpected data. Write access to unmapped memory has no effect.
[c] Three buffer pools span 96K total. The three pools must be a continuos allocation.
[d] PXE image is mapped into this region at boot. The host may access the PXE image at this location, 64K total memory is made available.
[e] The Buffer Pool-3 is not available in BCM5704C and BCM5704S MACs.

Table 83: BCM5705, BCM5788, BCM5721, BCM5751, and BCM5752 Address Map

Region	Size	NIC RISC Processor View	Host Flat View	Host Standard View	Host UNDI View
Unmapped	256B	0x00000000– 0x000000FF	0x01000000– 0x010000FF	0x00000000–[a] 0x000000FF	0x00000000–[a] 0x000000FF
Send ring RCB	16B	0x00000100– 0x0000010F	0x01000100– 0x0100010F	0x00000100–[a] 0x0000010F	0x00000100–[a] 0x0000010F
Unmapped	240B	0x00000110– 0x000001FF	0x01000110– 0x010001FF	0x00000110– 0x000001FF	0x00000110– 0x000001FF
Receive return ring RCB	16B	0x00000200– 0x0000020F	0x01000200– 0x0100020F	0x00000200–[a] 0x0000020F	0x00000200–[a] 0x0000020F
Unmapped	240B	0x00000210– 0x000002FF	0x01000210– 0x010002FF	0x00000210– 0x000002FF	0x00000210– 0x000002FF
Unmapped	2KB	0x00000300– 0x00000AFF	0x01000300– 0x01000AFF	0x00000300–[a] 0x00000AFF	0x00000300–[a] 0x00000AFF
Unmapped	80B	0x00000B00– 0x00000B4F	0x01000B14– 0x01000B4F	0x00000B14– 0x00000B4F	0x00000B14– 0x00000B4F
Software Gencomm	1 KB	0x00000B50– 0x00000F4F	0x01000B50– 0x01000F4F	0x00000B50–[a] 0x00000F4F	0x00000B50–[a] 0x00000F4F
Unmapped	4 KB	0x00000F50– 0x00001FFF	0x01000F50– 0x01001FFF	0x00000F50– 0x00001FFF	0x00000F50– 0x00001FFF
Unmapped	8 KB	0x00002000– 0x00003FFF	0x01002000– 0x01003FFF	0x00002000– 0x00003FFF	0x00002000– 0x00003FFF

Table 83: BCM5705, BCM5788, BCM5721, BCM5751, and BCM5752 Address Map (Cont.)

Region	*Size*	*NIC RISC Processor View*	*Host Flat View*	*Host Standard View*	*Host UNDI View*
Send ring	2 KB	0x00004000– 0x000047FF	0x01004000– 0x010047FF	0x00004000–[a] 0x000047FF	0x00004000–[a] 0x000047FF
Unmapped	6 KB	0x00004800– 0x00005FFF	0x01004800– 0x01005FFF	0x00004800– 0x00005FFF	0x00004800– 0x00005FFF
Standard receive ring	4 KB	0x00006000– 0x00006FFF	0x01006000– 0x01006FFF	0x00006000–[a] 0x00006FFF	0x00006000–[a] 0x00006FFF
Unmapped	4 KB	0x00007000– 0x00007FFF	0x01007000– 0x01007FFF	0x00007000– 0x00007FFF	0x00007000– 0x00007FFF
TXMBUF	8 KB	0x00008000– 0x00009FFF	0x01008000– 0x01009FFF	0x00008000–[a] 0x00009FFF	0x00008000–[a] 0x00009FFF
Unmapped	24 KB	0x0000A000– 0x0000FFFF	0x0100A000– 0x0100FFFF	0x0000A000– 0x0000FFFF	0x0000A000– 0x0000FFFF
RXMBUF/scratch pad (BCM5705, BCM5705M only)	56 KB	0x00010000– 0x0001DFFF	0x01010000– 0x0101DFFF	0x00010000–[a] 0x0001DFFF	0x00010000–[a] 0x0001DFFF
RXMBUF/scratch pad (BCM5721/ BCM5751 only)	64 KB	0x00010000– 0x0001FFFF	0x01010000– 0x0101FFFF	0x00010000–[a] 0x0001FFFF	0x00010000–[a] 0x0001FFFF
Unmapped (BCM5705, BCM5705M only)	15M+ 912 KB	0x0001E000– 0x00FFFFFF	0x0101E000– 0x01FFFFFF	0x0001E000– 0x00FFFFFF	0x0001E000– 0x00FFFFFF
Unmapped (BCM5721/BCM5751 only)	15M + 904 KB	0x00020000– 0x00FFFFFF	0x01020000– 0x00FFFFFF	0x00020000– 0x00FFFFFF	0x00020000– 0x00FFFFFF
Unmapped	16 KB	0x08000000– 0x08003FFF	–	–	–
RXCPU ROM	2 KB	0x40000000– 0x400007FF	–	–	–
PCI configuration	256B	0xC0000400– 0xC00000FF	0x00000000– 0x000000FF	0x00000000– 0x000000FF	0x00000000– 0x000000FF
High priority mailbox	512B	–	0x00000200– 0x000003FF	0x00000200– 0x000003FF	0x00005800– 0x000059FF[b]
Functional registers	31 KB	0xC0000400– 0xC0007FFF	0x00000400– 0x00007FFF	0x00000400– 0x00007FFF	0x00000100– 0x00007FFF[b]
Mailboxes	1 MB	–	0x00100000– 0x001FFFFF	–	–
Unmapped	32 KB	0xC0030000– 0xC0037FFF	0xC0030000– 0xC0037FFF	0xC0030000– 0xC0037FFF	0xC0030000– 0xC0037FFF
RXCPU ROM slave access	2 KB	0xC0038000– 0xC00387FF	0xC0038000– 0xC00387FF	0xC0038000– 0xC00387FF	0xC0038000– 0xC00387FF[b]

a. Indirect access via Memory Window Base Address and Memory Window Data registers pair.
b. Indirect Access via Register Window Base Address and Register Window Data registers pair.

Table 84: BCM5714C, BCM5714S, BCM5715C, BCM5715S Memory Map

Region	*Size*	*NIC CPU View*	*Host Flat View*	*Host Standard View*	*Host UNDI View*
Unmapped	256B	0x00000000–0x000000FF	0x01000000–0x010000FF	0x00000000–*0x000000FF	0x00000000–*0x000000FF
Send ring RCB	16B	0x00000100–0x0000010F	0x01000100–0x0100010F	0x00000100–*0x0000010F	0x00000100–*0x0000010F
Unmapped	240B	0x00000110–0x000001FF	0x01000110–0x010001FF	0x00000110–0x000001FF	0x00000110–0x000001FF
Receive return ring RCB	16B	0x00000200–0x0000020F	0x01000200–0x0100020F	0x00000200–*0x0000020F	0x00000200–*0x0000020F
Unmapped	240B	0x00000210–0x000002FF	0x01000210–0x010002FF	0x00000210–0x000002FF	0x00000210–0x000002FF
Unmapped	2 KB	0x00000300–0x00000AFF	0x01000300–0x01000AFF	0x00000300–*0x00000AFF	0x00000300–*0x00000AFF
Unmapped	80B	0x00000B00–0x00000B4F	0x01000B14–0x01000B4F	0x00000B14–0x00000B4F	0x00000B14–0x00000B4F
Software Gencomm	1 KB	0x00000B50–0x00000F4F	0x01000B50–0x01000F4F	0x00000B50–*0x00000F4F	0x00000B50–*0x00000F4F
Unmapped	4 KB	0x00000F50–0x00001FFF	0x01000F50–0x01001FFF	0x00000F50–0x00001FFF	0x00000F50–0x00001FFF
Unmapped	8 KB	0x00002000–0x00003FFF	0x01002000–0x01003FFF	0x00002000–0x00003FFF	0x00002000–0x00003FFF
Send ring	2 KB	0x00004000–0x000047FF	0x01004000–0x010047FF	0x00004000–*0x000047FF	0x00004000–*0x000047FF
Unmapped	6 KB	0x00004800–0x00005FFF	0x01004800–0x01005FFF	0x00004800–0x00005FFF	0x00004800–0x00005FFF
Standard receive ring	4 KB	0x00006000–0x00006FFF	0x01006000–0x01006FFF	0x00006000–*0x00006FFF	0x00006000–*0x00006FFF
Unmapped	4 KB	0x00007000–0x00007FFF	0x01007000–0x01007FFF	0x00007000–0x00007FFF	0x00007000–0x00007FFF
TXMBUF	22 KB	0x00008000–0x0000C7FF	0x01008000–0x0100C7FF	0x00008000–*0x0000C7FF	0x00008000–*0x0000C7FF
Unmapped	10 KB	0x0000D800–0x0000FFFF	0x0100D800–0x0100FFFF	0x0000D800–0x0000FFFF	0x0000D800–0x0000FFFF
RXMBUF	32 KB	0x00010000–0x00017FFF	0x01010000–0x01017FFF	0x00010000–*0x00017FFF	0x00010000–*0x00017FFF
Unmapped	15M+912 KB	0x00020000–0x00FFFFFF	0x01020000–0x01FFFFFF	0x00020000–0x00FFFFFF	0x00020000–0x00FFFFFF
RX-CPU scratch pad memory	36 KB	0x08000000–0x08008FFF	–	–	–
RXCPU ROM	2 KB	0x40000000–0x400007FF	–	–	–
PCI configuration	256B	0xC0000000–0xC00000FF	0x00000000–0x000000FF	0x00000000–0x000000FF	0x00000000–0x000000FF

Table 84: BCM5714C, BCM5714S, BCM5715C, BCM5715S Memory Map (Cont.)

Region	*Size*	*NIC CPU View*	*Host Flat View*	*Host Standard View*	*Host UNDI View*
High priority mailbox	512B	–	0x00000200– 0x000003FF	0x00000200– 0x000003FF	0x00005800–** 0x000059FF
Functional registers	31 KB	0xC0000400– 0xC0007FFF	0x00000400– 0x00007FFF	0x00000400– 0x00007FFF	0x00000100–** 0x00007FFF
Mailboxes	1 MB	–	0x00100000– 0x001FFFFF	–	–
Unmapped	32 KB	0xC0030000– 0xC0037FFF	0xC0030000– 0xC0037FFF	0xC0030000– 0xC0037FFF	0xC0030000– 0xC0037FFF
RXCPU SPAD slave access	0 KB to 32 KB	0xC0030000– 0xC0037FFF	0xC0030000–** 0xC0037FFF	0xC0030000–** 0xC0037FFF	0xC0030000–** 0xC0037FFF
RXCPU SPAD slave access	33 KB to 36 KB	0xC003C000– 0xC003CFFF	0xC003C000–** 0xC003CFFF	0xC003C000–** 0xC003CFFF	0xC003C000–** 0xC003CFFF
RXCPU ROM slave access	2 KB	0xC0038000– 0xC00387FF	0xC0038000–** 0xC00387FF	0xC0038000–** 0xC00387FF	0xC0038000–** 0xC00387FF

* Indirect Access via Memory Window Base Address and Memory Window Data register pair.
** Indirect Access via Register Window Base Address and Register Window Data register pair.

The BCM5700 MAC supports up to 16 MB of external synchronous static memory (SSRAM). A 512K SSRAM 64x64 is an example of a smaller memory chip that is available commercially. When this SSRAM is attached to the BCM5700 MAC, the maximum number of send rings may be increased from 4 to 16. Additionally, the mini receive producer ring may be enabled. Enabling these rings will displace the 96K allocated for packet buffers to an external SSRAM address range (see Table 85 on page 175).

The BCM5700 MAC memory interface supports up to 16M of external SSRAM. Software can allocate a maximum of 8M for buffer pools and 8M for firmware value adds. The hardware design must provide external memory for software to leverage extra features/capabilities in the MAC. The following configuration is possible when external SSRAM is used with BCM5700:

- Receive mini producer ring enabled
- Receive standard producer ring enabled
- Receive jumbo producer ring enabled
- Send rings 0–15 enabled
- 96K–8M for packet buffers
- 0-8M addressable memory for firmware value-adds

Table 85: BCM5700-(External SRAM Only) Memory Map

Region	*Size*	*NIC RISC Processor View*	*Host Flat View*	*Host Standard View*	*Host UNDI View*
Page zero	256B	0x00000000– 0x000000FF	0x01000000– 0x010000FF	0x00000000–[a] 0x000000FF	0x00000000–[a] 0x000000FF
Send ring RCB	256B	0x00000100– 0x000001FF	0x01000100– 0x010001FF	0x00000100–[a] 0x000001FF	0x00000100–[a] 0x000001FF
Receive return ring RCB	256B	0x00000200– 0x000002FF	0x01000200– 0x010002FF	0x00000200–[a] 0x000002FF	0x00000200–[a] 0x000002FF

Table 85: BCM5700-(External SRAM Only) Memory Map (Cont.)

Region	Size	NIC RISC Processor View	Host Flat View	Host Standard View	Host UNDI View
Statistics block	2 KB	0x00000300– 0x00000AFF	0x01000300– 0x01000AFF	0x00000300–[a] 0x00000AFF	0x00000300–[a] 0x00000AFF
Status block	80B	0x00000b00– 0x00000B4F	0x01000b00– 0x01000B4F	0x00000b00–[a] 0x00000B4F	0x00000b00–[a] 0x00000B4F
Software Gencomm	1 KB	0x00000b50– 0x00000FFF	0x01000b50– 0x01000FFF	0x00000b50–[a] 0x00000FFF	0x00000b50–[a] 0x00000FFF
Unmapped[b]	4 KB	0x00001000– 0x00001FFF	0x01001000– 0x01001FFF	0x00001000–[a] 0x00001FFF	0x00001000–[a] 0x00001FFF
DMA descriptors	8 KB	0x00002000– 0x00003FFF	0x01002000– 0x01003FFF	0x00002000–[a] 0x00003FFF	0x00002000–[a] 0x00003FFF
Send rings 1–4	8 KB	0x00004000– 0x00005FFF	0x01004000– 0x01005FFF	0x00004000–[a] 0x00005FFF	0x00004000–[a] 0x00005FFF
Send rings 5–6	4 KB	0x00006000– 0x00006FFF	0x01006000– 0x01006FFF	0x00006000–[a] 0x00006FFF	0x00006000–[a] 0x00006FFF
Send rings 7–8	4 KB	0x00007000– 0x00007FFF	0x01007000– 0x01007FFF	0x00007000–[a] 0x00007FFF	0x00007000–[a] 0x00007FFF
Send rings 9–16	16 KB	0x00008000– 0x0000BFFF	0x01008000– 0x0100BFFF	0x00008000–[a] 0x0000BFFF	0x00008000–[a] 0x0000BFFF
Standard receive rings	4 KB	0x0000c000– 0x0000CFFF	0x0100c000– 0x0100CFFF	0x0000c000–[a] 0x0000CFFF	0x0000c000–[a] 0x0000CFFF
Jumbo receive rings	4 KB	0x0000d000– 0x0000DFFF	0x0100d000– 0x0100DFFF	0x0000d000–[a] 0x0000DFFF	0x0000d000–[a] 0x0000DFFF
Mini receive rings	8 KB	0x0000e000– 0x0000FFFF	0x0100e000– 0x0100FFFF	0x0000e000–[a] 0x0000FFFF	0x0000e000–[a] 0x0000FFFF
Available region[c] or (expansion ROM)[d]	32 KB	0x00010000– 0x00017FFF	0x01010000– 0x01017FFF	0x00010000–[a] 0x00017FFF	0x00010000–[a] 0x00017FFF
Available region or (expansion ROM)	32 KB	0x00018000– 0x0001FFFF	0x01018000– 0x0101FFFF	0x00018000–[a] 0x0001FFFF	0x00018000–[a] 0x0001FFFF
External memory (buffer pool)[e]	16 MB	0x00020000– 0x00FFFFFF	0x01020000– 0x01FFFFFF	0x00020000–[a] 0x00FFFFFF	0x00020000–[a] 0x00FFFFFF

a. Indirect access via Memory Window Base Address and Memory Window Data registers pair.
b. Read access to unmapped memory returns unexpected data. Write access to unmapped memory has no effect.
c. This region is made available to firmware for value adds.
d. PXE image is mapped into this region at boot. The host may access the PXE image at this location, 64K total memory is made available
e. The external SSRAM bank is available for packet buffers. A minimum of 96K should be made available.

The Statistics Block Memory (see Table 54 on page 114) includes MAC statistics, interface statistics, and BCM57XX MIB statistics.

Host software should use NIC local memory offsets when initializing BCM57XX registers. The indirect, flat, and standard modes are for host processor access to MAC local resources. Host software must initialize the pool for internal buffer allocation. The Mbuf_Pool_Base_Address register (see "Buffer Manager Control Registers" on page 466) configures the base location where packet pool allocations are made. This location is different based on internal SSRAM or external SSRAM configurations (see the Buffer Pool memory offsets in Table 82 on page 171 and Table 85 on page 175). When send rings 4-15 are enabled, the entire buffer pool must be relocated to external SSRAM locations.

Note: The buffer pool cannot be split across internal and external memory.

Once a base address is configured, the size of the pool is specified by writing a value to the Mbuf_Pool_Length register (see "Buffer Manager Control Registers" on page 466). When an internal memory configuration is programmed, the pool length should be set to 96K. An external memory configuration allows software to allocate a maximum of 8M of buffers. Host software should determine the amount of external SSRAM provided in the hardware application, and then configure the pool length accordingly. Both the RX and TX datapaths allocate from the same buffer pool. The receive datapath always has allocation priority over the TX datapath. The TX datapath can wait to complete sends, whereas the RX datapath must drop packets, which is costly. External memory extends the buffer pools from both RX/TX operations and provides packet elasticity for bursty wire traffic.

The DMA pool must also be configured for BCM5700 through BCM5704 MACs. The DMA pool provides internal resources for the MAC to move buffer descriptors (BDs) and frame buffers from/to host memory. The RX/TX MAC will get allocations from the pool, so the Read/Write DMA engines can complete a data transfer operation, over the PCI bus. The DMA_Descriptor_Pool_Base_Address (see "Buffer Manager Control Registers" on page 466) should be configured using the memory locations specified in Table 82 on page 171 and Table 85 on page 175. Hardware applications using either internal or external SSRAM should configure this register to the value 0x2000. The DMA Descriptor Pool Length register (see "Buffer Manager Control Registers" on page 466) should be configured to span the entire DMA descriptor memory map of 8K. The BCM5705, BCM5788, BCM5721, BCM5751, BCM5714C, BCM5714S, BCM5715C, BCM5715S, and BCM5752 MACs do not have a DMA Descriptor resource.

The Statistics_Base_Address register (see "Statistics Base Address Register (Offset 0x3C40)" on page 457) should be set to 0x300 on MAC devices that support Statistics memory block. The Status_Block_Base_Address register (see "Status Block Base Address Register (Offset 0x3C44)" on page 457) should be set to 0xB00 in all NetXtreme BCM57XX devices. The configuration of these registers is independent of internal and external SSRAM. Reference the memory maps in Table 82 on page 171 and Table 85 on page 175. The BCM5705, BCM5788, BCM5721, and BCM5751, BCM5714C, BCM5714S, BCM5715C, BCM5715S, and BCM5752 MACs do not have Statistics memory block.

Section 9: PCI

CONFIGURATION SPACE

DESCRIPTION

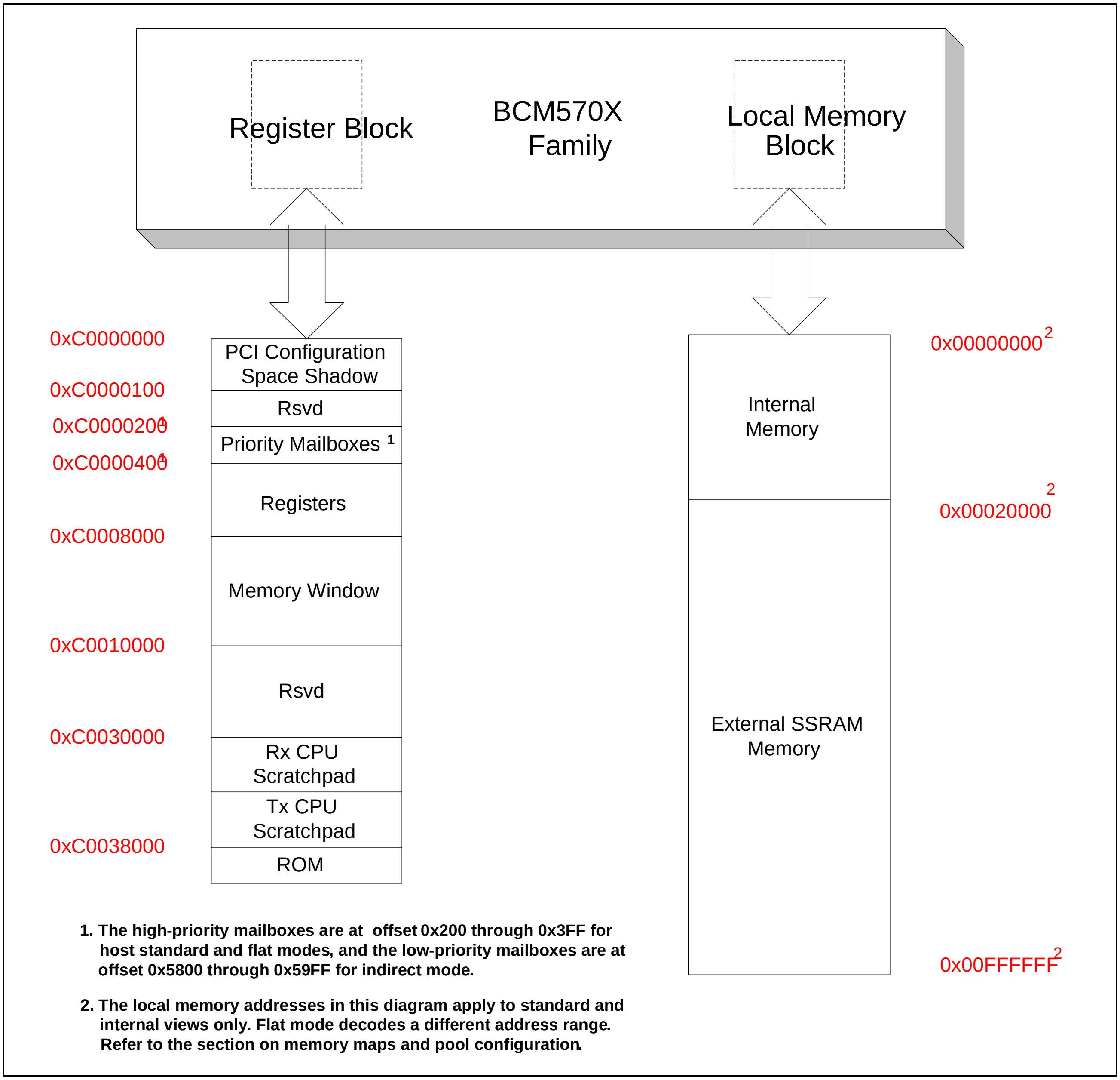
PCI, PCI-X, and PCIe devices must implement sixteen 32-bit PCI registers. These registers are required for a device to have PCI compliance. The format and layout of these registers is defined in the PCI 2.2 specification. Capability registers provide system BIOS and Operating Systems visibility into a set of optional features, which devices may implement. Although the capability registers are not required, the structure and mechanism for chaining auxiliary capabilities is defined in the PCI specification. Both software and BIOS must implement algorithms to fetch and program capabilities fields accordingly. Refer to section 6.7 of the PCI SIIG 2.2 specification. Additional PCI configuration space may be used for device-specific registers. However, device-specific registers are not exposed to system software, according to a specification/standard. System software cannot probe device specific registers without a predetermined understanding of the device and its functionality. In summary, three types of PCI configuration space registers may be exposed by any particular device:

- Required
- Optional capabilities
- Device specific

Network devices implement large quantities of registers, and these registers could consume huge amounts of PCI configuration space. PCI configuration access is not very efficient, on a performance basis. For example, Intel x86 architectures use two I/O mapped I/O addresses 0xCF8 and 0xCFC for host-based access to PCI configuration space. Should a host device driver access these I/O addresses on every device read/write, CPU overhead would grow greatly. Generally, host device drivers should not use PCI configuration space for standard I/O and control programming. There is one special case—Universal Network Device Interface (UNDI) drivers. UNDI drivers may not have access to host memory mapped registers when operating in real-mode; thus, an indirect mode of access is necessary. The BCM57XX family implements a PCI indirect mode for memory, registers, and mailboxes access. A specific example of a device driver, which uses indirect mode, is the Preboot Execution (PXE) driver. PXE drivers may be stored in either option ROMs or directly in the system BIOS.

Most host device drivers use register blocks, which are mapped into host memory. Memory Mapped I/O is an efficient mechanism for PCI devices to use system resources. The type and extent of this memory mapping depends upon the MAC's configuration (see the operational characteristics subsection). A typical PCI device will decode a range of physical (bus) addresses, which do not conflict with physical memory or other PCI devices. Each device on the PCI bus will request a range of physical memory, and the PnP BIOS will assign mutually exclusive resources to that device. The size and range of resource is based upon each device's hardwired programming of the BAR. The BCM57XX family implements two modes of memory mapped I/O—Standard and Flat. I/O mapped I/O is not supported by the BCM57XX family, and there are no I/O space registers.

Two programmable blocks expose BCM57XX family functionality to host software. The first is a register block. The second is a memory block. The register and memory blocks map into address spaces, based on processor context. For example, the BCM57XX family has two internal RISC processors. Both RISC processors have an internal view of the register and memory blocks. This view is one large contiguous and addressable range, where the register block maps starting at offset 0xC0000000. On the other hand, host processors have two entirely different views. When the BCM57XX family is configured in standard mode, the register block is mapped into a 64K host memory range. The host processor must use a memory window or indirect mode to access the memory block. A Flat mode configuration maps both the memory and register blocks into 32 MB of address space. Flat mode ties up a much larger range of host memory addresses. It is fundamental to understand that the register and memory blocks are not necessarily tied together. The PCI mode and processor context all affect how software views both blocks (see the following figure).



1. The high-priority mailboxes are at offset 0x200 through 0x3FF for host standard and flat modes, and the low-priority mailboxes are at offset 0x5800 through 0x59FF for indirect mode.
2. The local memory addresses in this diagram apply to standard and internal views only. Flat mode decodes a different address range. Refer to the section on memory maps and pool configuration.

Figure 63: Local Contexts

The following components are involved in BCM57XX configuration space mapping:

- Base Address registers
- Standard Mode Map Mode
- Flat Memory Map Mode
- Indirect Access Mode
- Configuration Space Header
- Host Memory
- MAC registers
- MAC Local Memory

Functional Overview

PCI Configuration Space Registers

The BCM57XX configuration space can be broken into two regions: Header and Device Specific. Table 134: "PCI Configuration Register Summary," on page 298 shows the registers implemented to support both PCI and PCI-X functionality in the BCM57XX family. Reserved fields in PCI configuration registers will always return zero.

PCI Required Header Region

The BCM57XX chips are single function network devices—these chips contain a Header Type register (see Figure 64 and "Header Type Register (Offset 0x0E)" on page 306), which is part of the required PCI configuration space. The header type register identifies single or multifunction devices by exposing a read-only status bit. Bit_7 in the header space record is de-asserted/cleared for the single function BCM57XX chips, since only one PCI function is exposed. The BCM57XX dual function devices like BCM5704, BCM 5714, and BCM5715 will have this bit set.

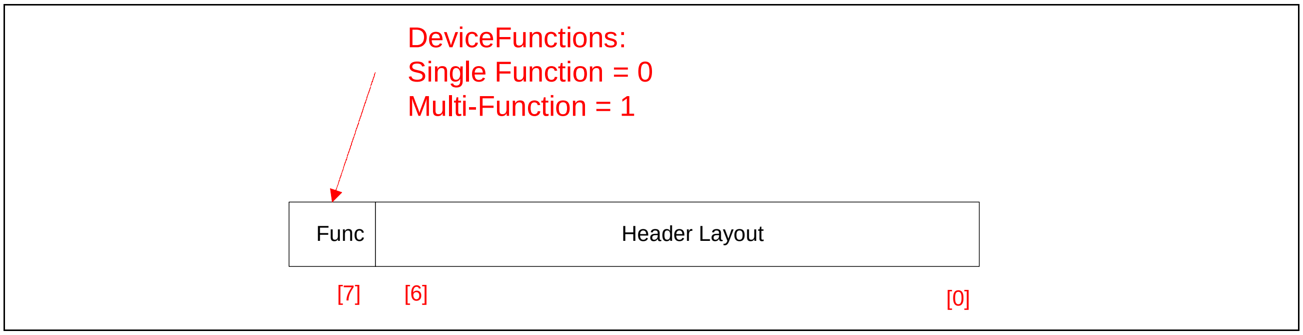


Figure 64: Header Type Register 0xE

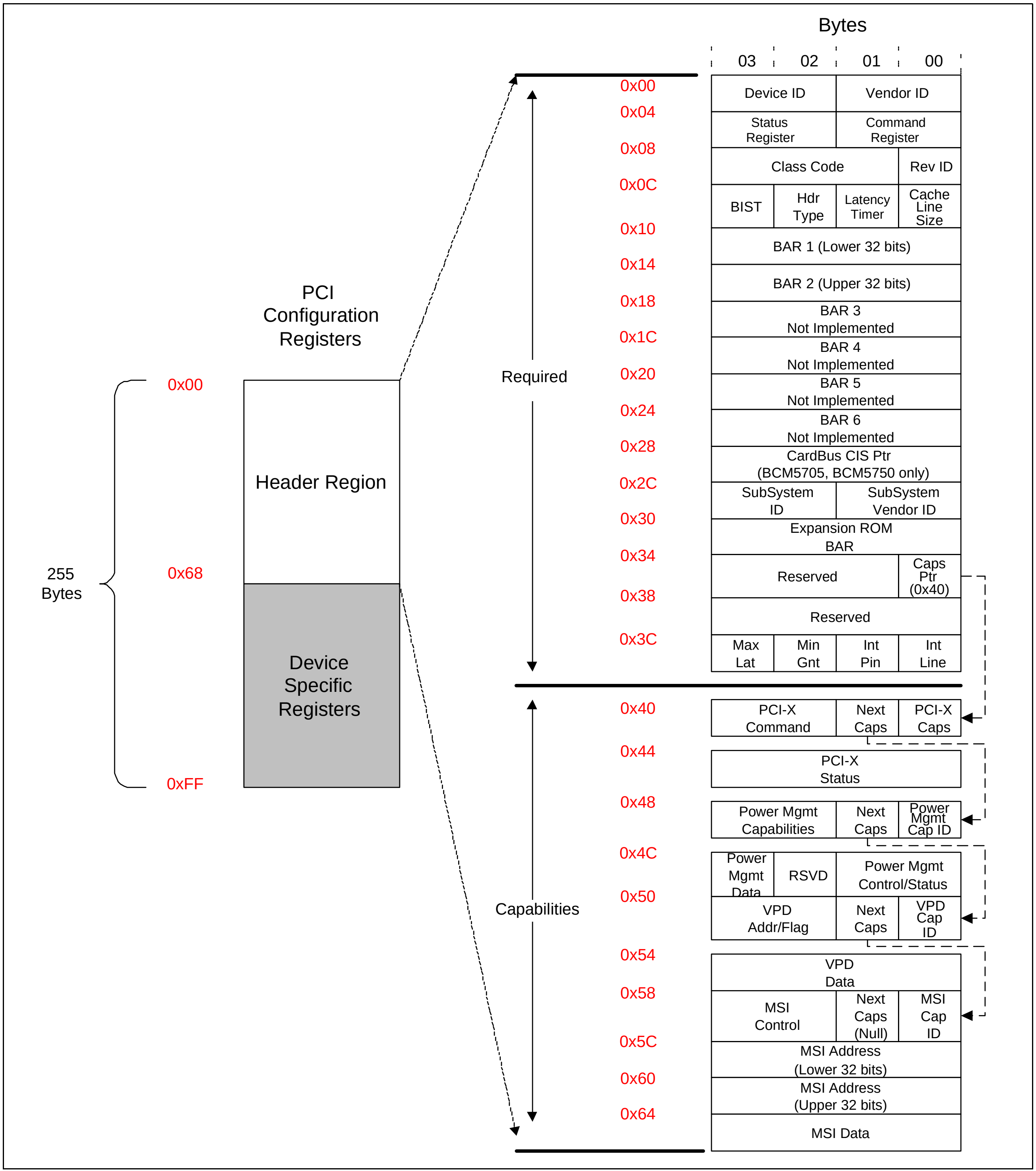
Note: BIOS programmers should take special care to read bit_7 in PCI Header Type register 0xE before scanning the BCM57XX PCI configuration space.

Single function PCI devices may decode access to non-implemented device functions in two ways, per Section 3.2.2.3.4 of the PCI 2.2 specification:

- A single function device may optionally respond to all function numbers as the same.
- May decode the function number field and respond only to function 0.

The BCM57XX single function chips follow the stated technique #1– BIOS code scanning multifunctions will get target response from function(s) 1-7, but these functions are essentially Shadows of function 0. Software that programs to function(s) 1-7 is re-mapped to function 0.

The header region (see Figure 65) is required by the PCI 2.2 specification. These registers must be implemented. The capabilities registers are optional; however, they must adhere to section 6.7 of the PCI SIIG 2.2 specification. Each capability has a unique ID, which is well-defined. The capabilities are chained using the Next Caps field, in the capability register. The last capability will have a Next Caps field, which is zeroed.



	03	02	01	00
0x00	Device ID		Vendor ID	
0x04	Status Register		Command Register	
0x08	Class Code			Rev ID
0x0C	BIST	Hdr Type	Latency Timer	Cache Line Size
0x10	BAR 1 (Lower 32 bits)			
0x14	BAR 2 (Upper 32 bits)			
0x18	BAR 3 Not Implemented			
0x1C	BAR 4 Not Implemented			
0x20	BAR 5 Not Implemented			
0x24	BAR 6 Not Implemented			
0x28	CardBus CIS Ptr (BCM5705, BCM5750 only)			
0x2C	SubSystem ID		SubSystem Vendor ID	
0x30	Expansion ROM BAR			
0x34	Reserved			Caps Ptr (0x40)
0x38	Reserved			
0x3C	Max Lat	Min Gnt	Int Pin	Int Line
0x40	PCI-X Command		Next Caps	PCI-X Caps
0x44	PCI-X Status			
0x48	Power Mgmt Capabilities		Next Caps	Power Mgmt Cap ID
0x4C	Power Mgmt Data	RSVD	Power Mgmt Control/Status	
0x50	VPD Addr/Flag		Next Caps	VPD Cap ID
0x54	VPD Data			
0x58	MSI Control		Next Caps (Null)	MSI Cap ID
0x5C	MSI Address (Lower 32 bits)			
0x60	MSI Address (Upper 32 bits)			
0x64	MSI Data			

Figure 65: Header Region Registers

The Header Region registers are shown in Table 86.

Table 86: Header Region Registers

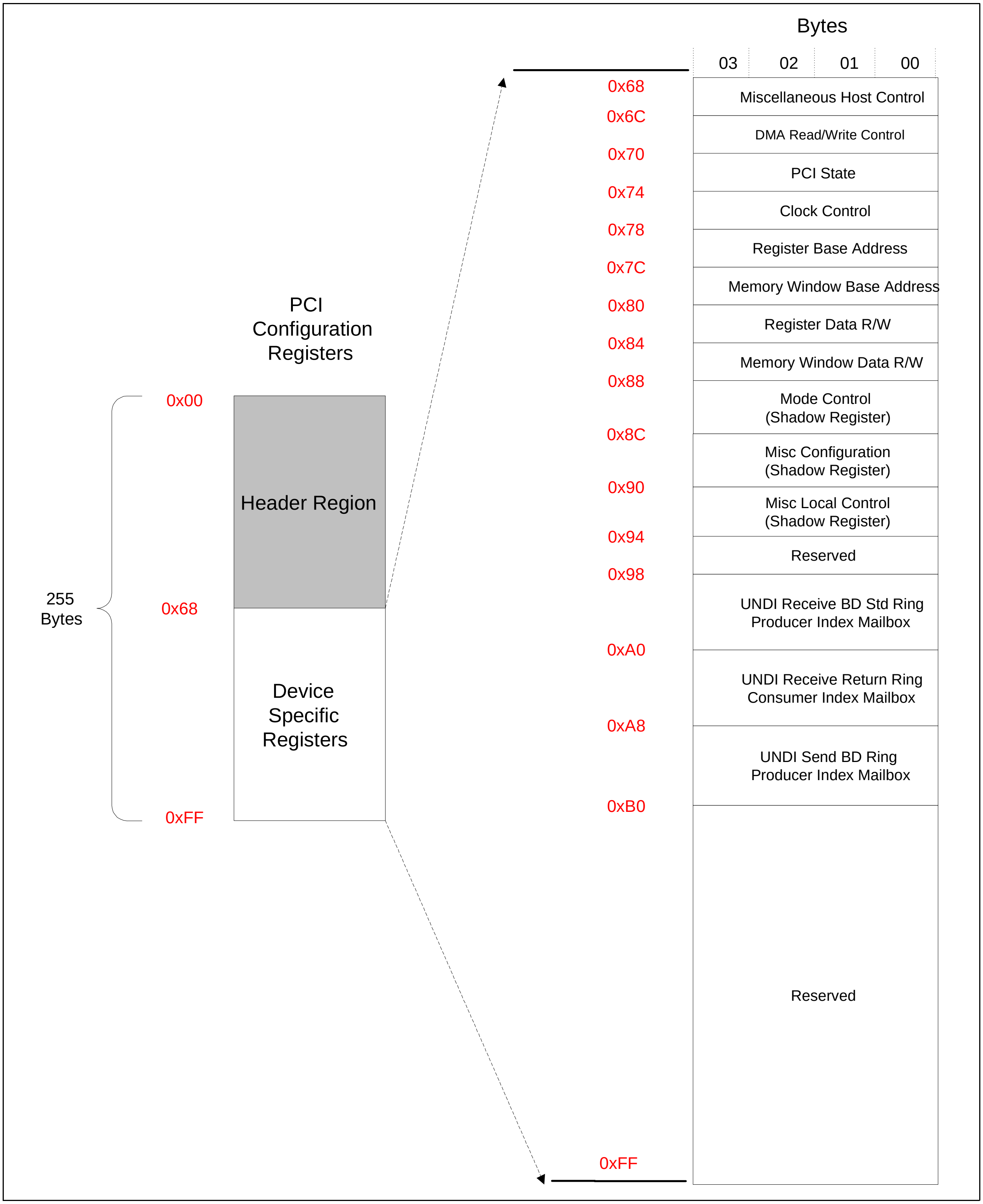
Register	***Cross Reference***
Vendor ID register	See "Vendor ID Register (Offset 0x00)" on page 300.
Device ID register	See "Device ID Register (Offset 0x02)" on page 301.
Command register	See "Command Register (Offset 0x04)" on page 302.
Status register	See "Status Register (Offset 0x06)" on page 303.
Revision ID register	See "Revision ID Register (Offset 0x08)" on page 304.
Class Code register	See "Class Code Register (Offset 0x09)" on page 305.
Cache Line Size register	See "Cache Line Size Register (Offset 0x0C)" on page 305.
Latency Timer register	See "Latency Timer Register (Offset 0x0D)" on page 305.
Header Type register	See "Header Type Register (Offset 0x0E)" on page 306.
BIST register	See "BIST Register (Offset 0x0F)" on page 306.
Base Address Register 1/2 register	See "Base Address Register 1/2 Register (Offset 0x10-0x17)" on page 306.
Subsystem Vendor ID register	See "Subsystem Vendor ID Register (Offset 0x2C)" on page 307.
Subsystem ID register	See "Subsystem ID Register (Offset 0x2E)" on page 309.
Expansion ROM Base Address register	See "Expansion ROM Base Address Register (Offset 0x30)" on page 310.
Capabilities Pointer register	See "Capabilities Pointer Register (Offset 0x34)" on page 310.
Interrupt Line register	See "Interrupt Line Register (Offset 0x3C)" on page 310.
Interrupt Pin register	See "Interrupt Pin Register (Offset 0x3D)" on page 311.
Minimum Grant register	See "Minimum Grant Register (Offset 0x3E)" on page 311.
Maximum Latency register	See "Maximum Latency Register (Offset 0x3F)" on page 311.
PCI-X Capability ID register	See "PCI-X Capability ID Register (Offset 0x40)" on page 312.
PCI-X Next Capabilities Pointer register	See "PCI-X Next Capabilities Pointer Register (Offset 0x41)" on page 312.
PCI-X Command register	See "PCI-X Command Register (Offset 0x42)" on page 312.
PCI-X Status register	See "PCI-X Status Register (Offset 0x44)" on page 314.
Power Management Capability ID register	See "Power Management Capability ID Register (Offset 0x48)" on page 316.
PM Next Capabilities Pointer register	See "PM Next Capabilities Pointer Register (Offset 0x49)" on page 316.
Power Management Capabilities (PMC) register	See "Power Management Capabilities Register (Offset 0x4A)" on page 317.
Power Management Control/Status (PMCSR) register	See "Power Management Control/Status Register (Offset 0x4C)" on page 318.
PMCSR-BSE register	See "PMCSR-BSE Register (Offset 0x4E)" on page 318.
Power Management Data register	See "Power Management Data Register (Offset 0x4F)" on page 319.
VPD Capability ID register	See "VPD Capability ID Register (Offset 0x50)" on page 320.
VPD Next Capabilities (MSI) Pointer register	See "VPD Next Capabilities Pointer Register (Offset 0x51)" on page 320.
VPD Flag and Address register	See "VPD Flag and Address Register (Offset 0x52)" on page 320.
VPD Data register	See "VPD Data Register (Offset 0x54)" on page 321.
MSI Capability ID register	See "MSI Capability ID Register (Offset 0x58)" on page 321.

Table 86: Header Region Registers (Cont.)

Register	*Cross Reference*
MSI Next Capabilities (NULL) Pointer register	See "MSI Next Capabilities Pointer Register (Offset 0x59)" on page 321.
Message Control register	See "Message Control Register (Offset 0x5A)" on page 322.
Message Address register	See "Message Address Register (Offset 0x5C)" on page 323.
Message Data register	See "Message Data Register (Offset 0x64)" on page 323.

PCI Device-Specific Region

Device-specific registers are not defined in the PCI 2.2 specification and are exactly as the name implies—specific to the BCM57XX family. These registers may be used by host software to configure or change the operational state of the MAC. The most notable feature exposed via the Device Specific registers is Indirect Mode. Host or system software may use Indirect Mode to access BCM57XX local memory and register space; no memory mapped I/O is necessary in indirect mode.



	03	02	01	00
0x68	Miscellaneous Host Control			
0x6C	DMA Read/Write Control			
0x70	PCI State			
0x74	Clock Control			
0x78	Register Base Address			
0x7C	Memory Window Base Address			
0x80	Register Data R/W			
0x84	Memory Window Data R/W			
0x88	Mode Control (Shadow Register)			
0x8C	Misc Configuration (Shadow Register)			
0x90	Misc Local Control (Shadow Register)			
0x94	Reserved			
0x98	UNDI Receive BD Std Ring Producer Index Mailbox			
0xA0	UNDI Receive Return Ring Consumer Index Mailbox			
0xA8	UNDI Send BD Ring Producer Index Mailbox			
0xB0	Reserved			
0xFF				

Figure 66: Device-Specific Registers

The Device-Specific registers are shown in the following table.

Table 87: Device-Specific Registers

Register	*Cross Reference*
Miscellaneous Host Control register	See "Miscellaneous Host Control Register (Offset 0x68)" on page 325.
DMA Read/Write Control register	See "DMA Read/Write Control Register (Offset 0x6C)" on page 327.
PCI State register	See "PCI State Register (Offset 0x70)" on page 332.
Register Base Address register	See "Register Base Address Register (Offset 0x78)" on page 340.
Memory Window Base Address register	See "Memory Window Base Address Register (Offset 0x7C)" on page 341.
Register Data register	See "Register Data Register (Offset 0x80)" on page 342.
Memory Window Data register	See "Memory Window Data Register (Offset 0x84)" on page 342.
UNDI Receive BD Standard Producer Ring Producer Index Mailbox	See "UNDI Receive BD Standard Producer Ring Producer Index Mailbox (Offset 0x98)" on page 345.
UNDI Receive Return Ring Consumer Index Mailbox	See "UNDI Receive Return Ring Consumer Index Mailbox (Offset 0xA0)" on page 345.
UNDI Send BD Producer Index Mailbox	See "UNDI Send BD Producer Index Mailbox (Offset 0xA8)" on page 345.

Indirect Mode

Host software may use indirect mode to access the BCM57XX resources, without using Memory Mapped I/O. Indirect mode shadows MAC resources to PCI configuration space registers. These shadow registers can be read/written by system software through PCI configuration space registers. The BCM57XX indirect mode registers expose the following MAC resources:

- Registers
- Local Memory
- Mailboxes

Indirect mode access can be used in conjunction with Standard or Flat Mode PCI access. Indirect mode has no interdependency on other PCI access modes and is a mode in itself.

Note: Host software must assert the Indirect_Mode_Access bit in the Miscellaneous Host Control register (see "Miscellaneous Host Control Register (Offset 0x68)" on page 325 to enable indirect mode.

Indirect Register Access

Two PCI configuration space register pairs give host software access to the BCM57XX register block. The Register_Base_Address register creates a position in the MAC register block. Valid positions range from 0x0000-0x8000 and 0x30000-0x38800 ranges. Access to the register block from 0x8000-0x30000, should be avoided and is not necessary. The Flat and Standard Modes do map a memory window into the 0x8000-0xFFFF ranges; however, the Memory Indirection register pair provides a more efficient mechanism to access the BCM57XX memory block. The Register_Data register allows host software to read/write, from the indirection position. The Register_Base_Address register can be perceived as creating a cursor/pointer into the register block. The Register_Data register allows host software to read/write to the location, specified by the Register_Base_Address. This register pair accesses the BCM57XX register block (see Figure 67).

Note: If indirect register access is performed using memory write cycles (i.e., by accessing the Register_Base_Address and Register_Data registers through memory mapped by the PCI BAR register), as opposed to PCI configuration write cycles, the host software must insert a read command to the Register_Base_Address register between two consecutive writes to the Register_Base_Address and Register_Data registers.

Note: The PCI BAR 0 register is only reset to 0 after a hard reset; otherwise, it maintains its value over GRC and PCI resets.

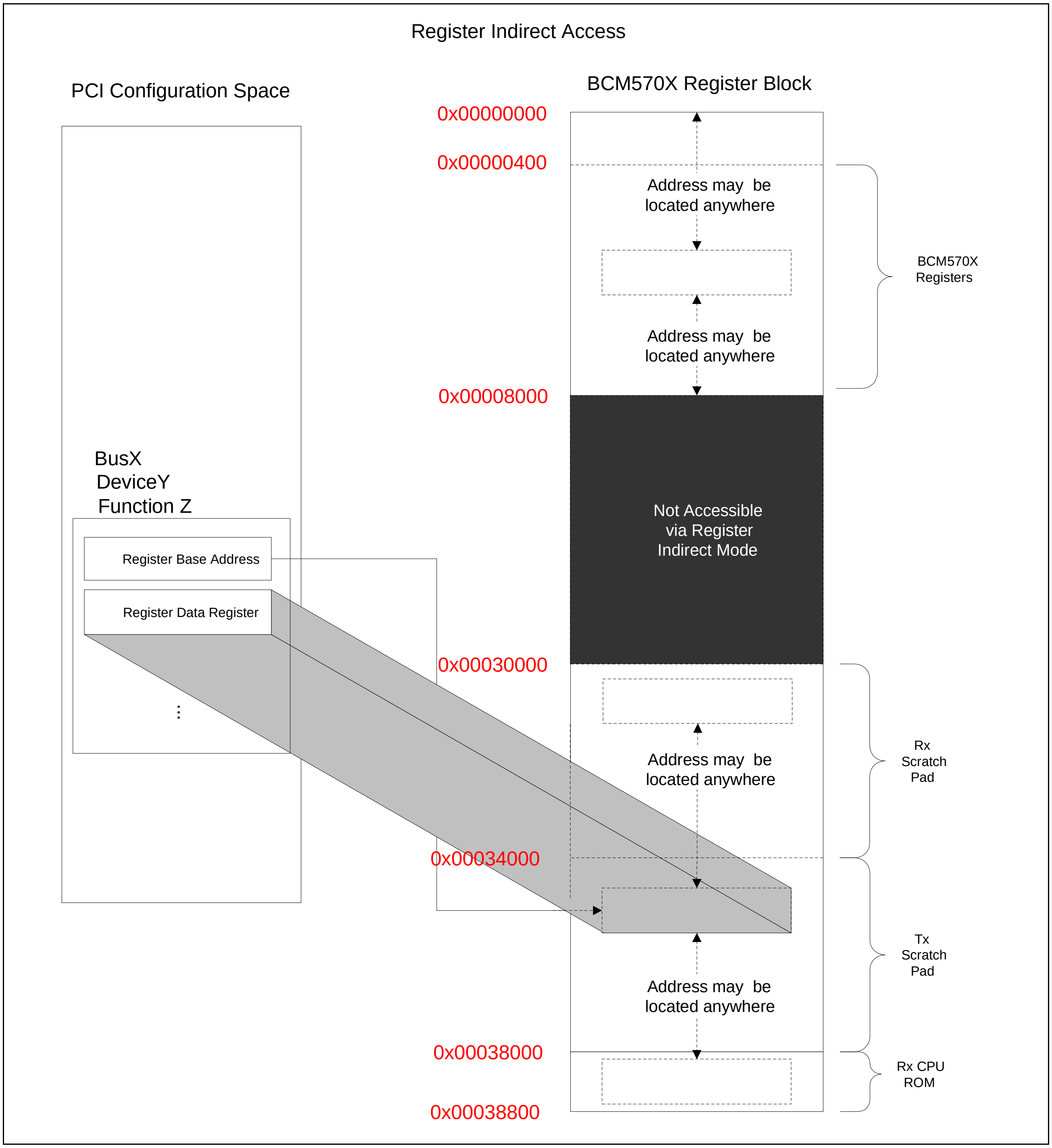


Figure 67: Register Indirect Access

Indirect Memory Access

Memory indirect mode operates in the same fashion to register indirect mode. There is a PCI configuration space register pair, which is used to access the BCM57XX memory block. The Memory_Window_Base_Address register positions a pointer/cursor in the local memory block. Unlike the Register_Base_Address register, the Memory_Window_Base_Address register may position at any valid offset. Access to ranges 0x00000-0x1FFFF (internal memory) and 0x20000-0xFFFFFF (external memory) is allowable. The obvious restriction on external memory access is both size and presence of external SSRAM (only the BCM5700 MAC supports the presence of external SSRAM). The Memory_Window_Data register is the read/write porthole for host software, using the previously positioned pointer/cursor. This register pair accesses the BCM57XX local memory block (see Figure 68).

Note: If Indirect Memory Access is performed using memory write cycles (i.e., by accessing the Memory_Window_Base_Address and Memory_Window_Data registers through memory mapped by the PCI BAR register), as opposed to PCI configuration write cycles, the host software must insert a read command to the Memory_Window_Base_Address register between two consecutive writes to the Memory_Window_Base_Address and Memory_Window_Data registers.

Note: The PCI BAR 0 register is only reset to 0 after a hard reset; otherwise, it maintains its value over GRC and PCI resets.

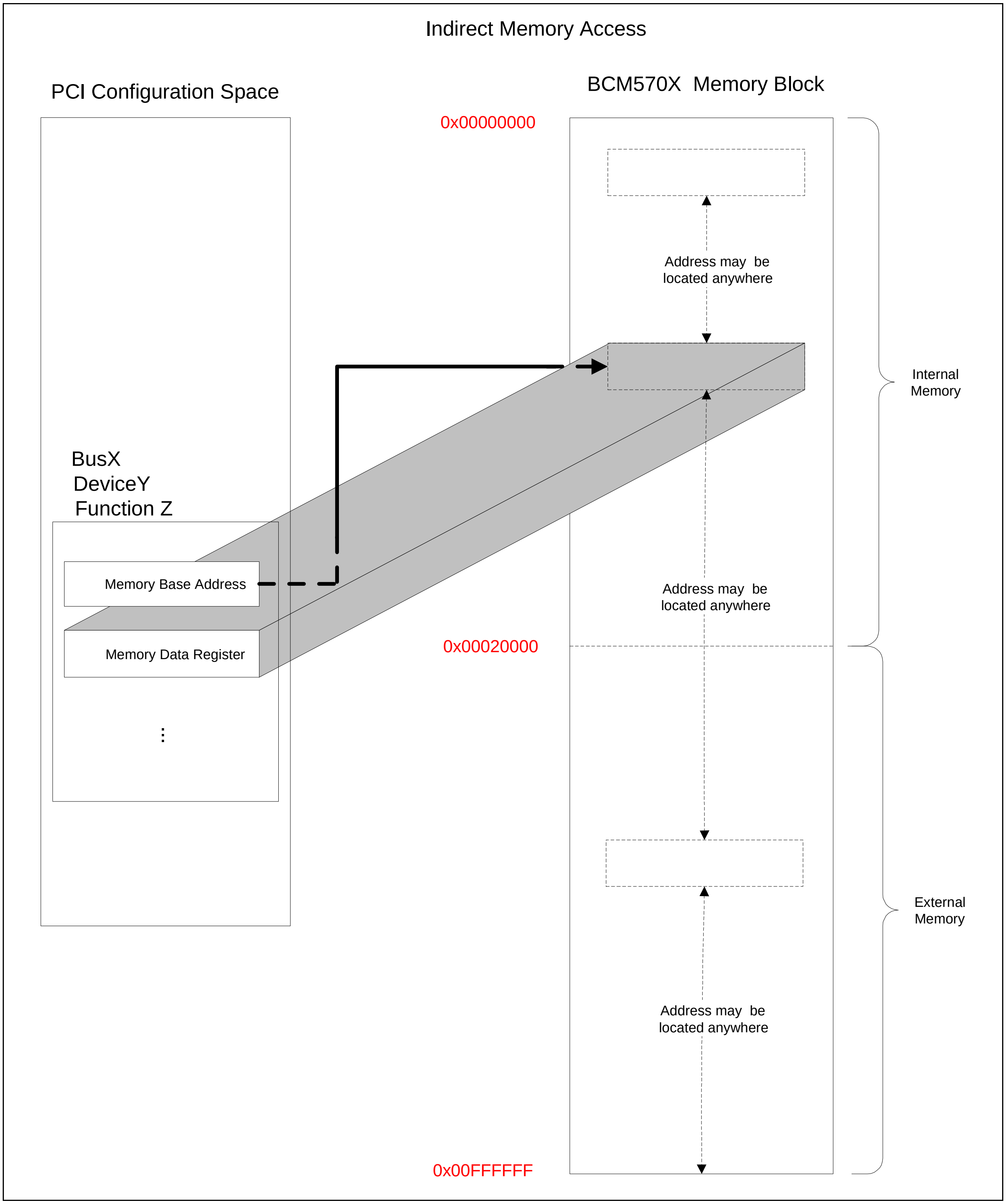


Figure 68: Indirect Memory Access

UNDI Mailbox Access

The UNDI mailboxes are shadows of three BCM57XX mailbox registers. All three mailboxes reside in the BCM57XX register block, not memory block. Unlike register and memory indirect access, the UNDI Mailboxes shadows are mapped 1:1 to a BCM57XX register; these shadow registers do not have an address register.

- The UNDI_RX_BD_Standard_Ring_Producer_Index_Mailbox register shadows a mailbox located at offset 0x5868 (see "Receive BD Standard Producer Ring Index Register (Offset 0x5868)" on page 492), in the BCM57XX register block. Any index update (write) to the UNDI_RX_BD_Standard_Ring_Producer_Index_Mailbox will advance the standard producer ring index; software signals hardware that an RX buffer descriptor is available.
- The UNDI_RX_BD_Return_Ring_Consumer_Index_Mailbox register corresponds to a mailbox located at offset 0x5880 (see "Receive BD Return Ring 1-16 Consumer Indices Registers (Offset 0x5880-0x58F8)" on page 492). A update (write) to this register indicates that host software has consumed a RX buffer descriptor(s); return rings contain filled Enet frames, from the receive MAC.
- Finally, the UNDI_TX_BD_NIC_Producer_Mailbox register maps to register offset 0x5980 in the BCM57XX register block (see "Send BD Ring 1-16 NIC Producer Indices Registers (Offset 0x5980-0x59F8, BCM5700 and BCM5701 Only)" on page 493). Host software writes to this register when Ethernet frame(s) are ready to be transmitted. Host software writes the index of buffer descriptor, which is ready for transmission.

Notice that all these UNDI shadows are the first or primary ring and not all the rings are shadowed into PCI configuration space. For example, Receive Return rings 2-16 do not have shadow registers. UNDI drivers only require a minimal set of registers to provide basic network connectivity. Functionality is the most important consideration. Fifteen additional receive return rings would extend the size of the Device Specific portion of the PCI Configuration Space registers.

The UNDI shadow registers alias three registers in the BCM57XX register block (see Figure 69).

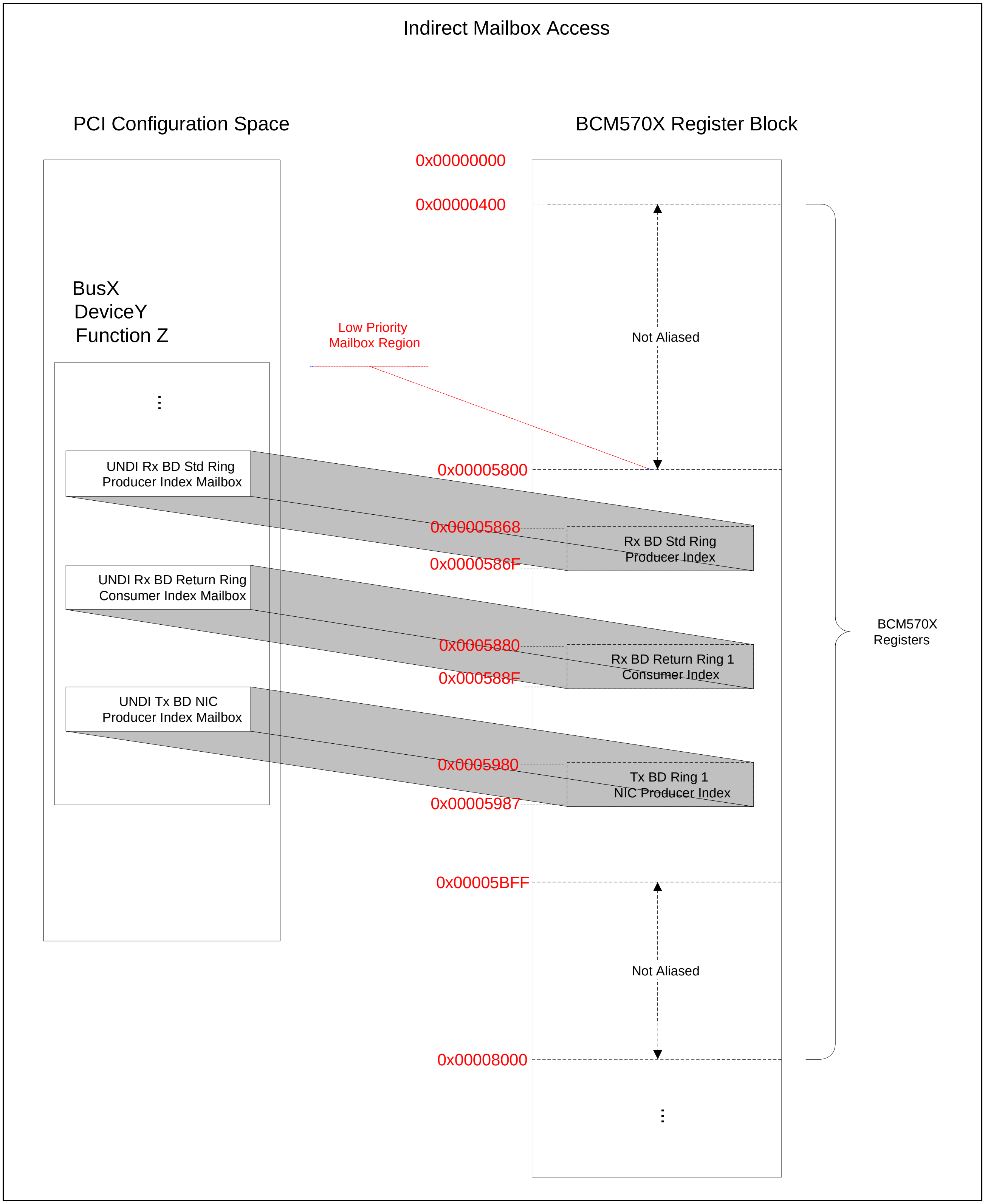

Indirect Mailbox Access
PCI Configuration Space
BCM570X Register Block
BusX
DeviceY
Function Z
⋮
UNDI Rx BD Std Ring Producer Index Mailbox
UNDI Rx BD Return Ring Consumer Index Mailbox
UNDI Tx BD NIC Producer Index Mailbox
Low Priority Mailbox Region
0x00000000
0x00000400
Not Aliased
0x00005800
0x00005868
0x0000586F
Rx BD Std Ring Producer Index
0x0005880
0x000588F
Rx BD Return Ring 1 Consumer Index
0x0005980
0x00005987
Tx BD Ring 1 NIC Producer Index
0x00005BFF
Not Aliased
0x00008000
⋮
BCM570X Registers

Figure 69: Low-Priority Mailbox Access for Indirect Mode

Standard Mode

Standard mode is the most useful memory mapped I/O view provided by the BCM57XX family (see Figure 70). 64K of host memory space must be made available. The PnP BIOS or OS will program BAR0 and BAR1 with a base address where the 64K address region may be decoded. The BAR registers point to the beginning of the host memory mapped regions where BCM57XX family can be accessed.

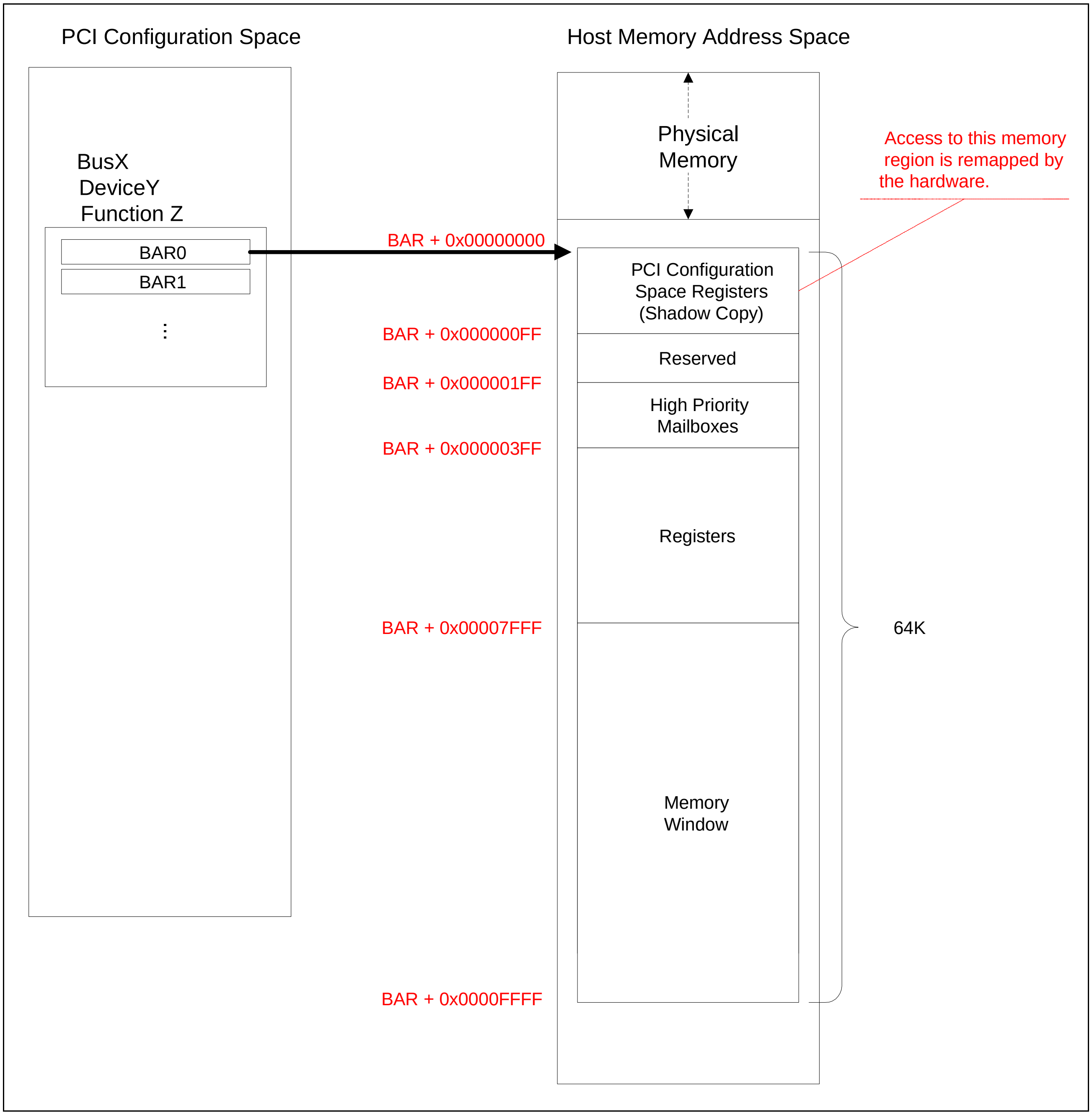


Figure 70: Standard Memory Mapped I/O Mode

The BCM57XX family resources listed in the following are decoded in the 64K address block.

Table 88: PCI Address Map Standard View

Offset	Name	Size
0x00000000-0x000000ff	PCI Configuration space	256 bytes
0x00000200-0x000003ff	High Priority Mailboxes	512 bytes
0x00000400-0x00007fff	BCM57XX registers	31 KB
0x00008000-0x0000ffff	Memory Window	32 KB

32K is partitioned for MAC control registers and 32K available for a memory access window. Range 0x0000-0x00FF is a complete shadow of the PCI configuration space registers—host software can also read/write to the BCM57XX family's PCI configuration space registers via the host memory map. Host software may use the shadow registers to change PCI register contents and avoid PCI configuration cycles (transactions). Again, using the host memory map is slightly more efficient. The MAC's control/status registers are mapped from 0x0400-0x8000. See "Register Definitions" on page 298 for complete register and bit definitions. Finally, the memory window range is 0x8000-0xFFFF. This 32K window is set in the PCI Configuration space using the Memory_Window_Base_Address register (see Figure 71). Bits 23-15 set the window aperture and bits 14-2 are effectively ignored/masked off. Bits 14-2 are relevant when host software uses memory indirection and the Memory_Window_Data register.

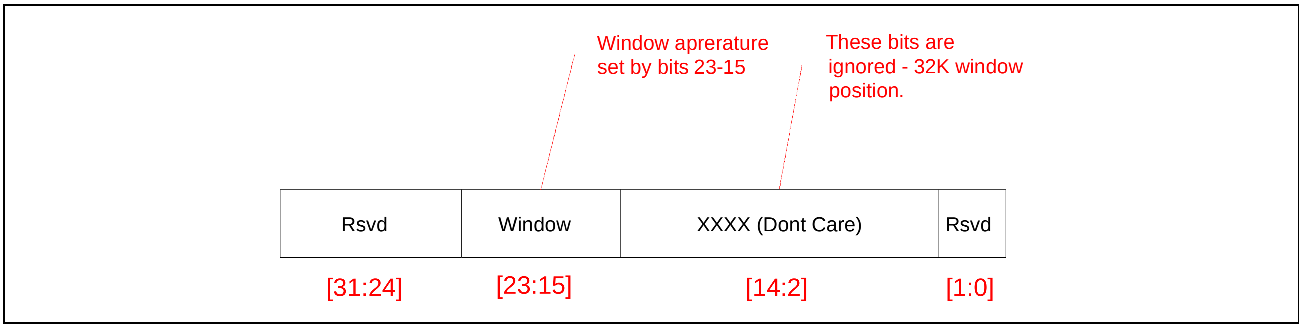

Window aprerature set by bits 23-15
These bits are ignored - 32K window position.
Rsvd
Window
XXXX (Dont Care)
Rsvd
[31:24]
[23:15]
[14:2]
[1:0]

Figure 71: Memory Window Base Address Register

Figure 72 shows how the 32K window can float in BCM57XX local memory. The window aligns on 32K boundaries. For example, the memory window may start on the following addresses: 0x8000, 0x10000, and 0x18000. The window aperture may be positioned in the internal memory range 0x00000000 to 0x0001FFFF. If external SSRAM is present, the window aperture may be position in the external memory range 0x00020000 to 0x00FFFFFF. When host software reads/writes to PCI_BAR + 32K + OFFSET in the host memory space, the BCM57XX family translates this read/write access to Memory_Window_Base_Address + OFFSET. Host software must not read/write from any address greater than PCI_BAR + 64K, since this memory space is not decoded by the BCM57XX family. Such an access may be decoded by another device, or simply go unclaimed on the PCI bus. Figure 72 shows the relationship between the Memory_Window_Base_Address register and the Memory Window.

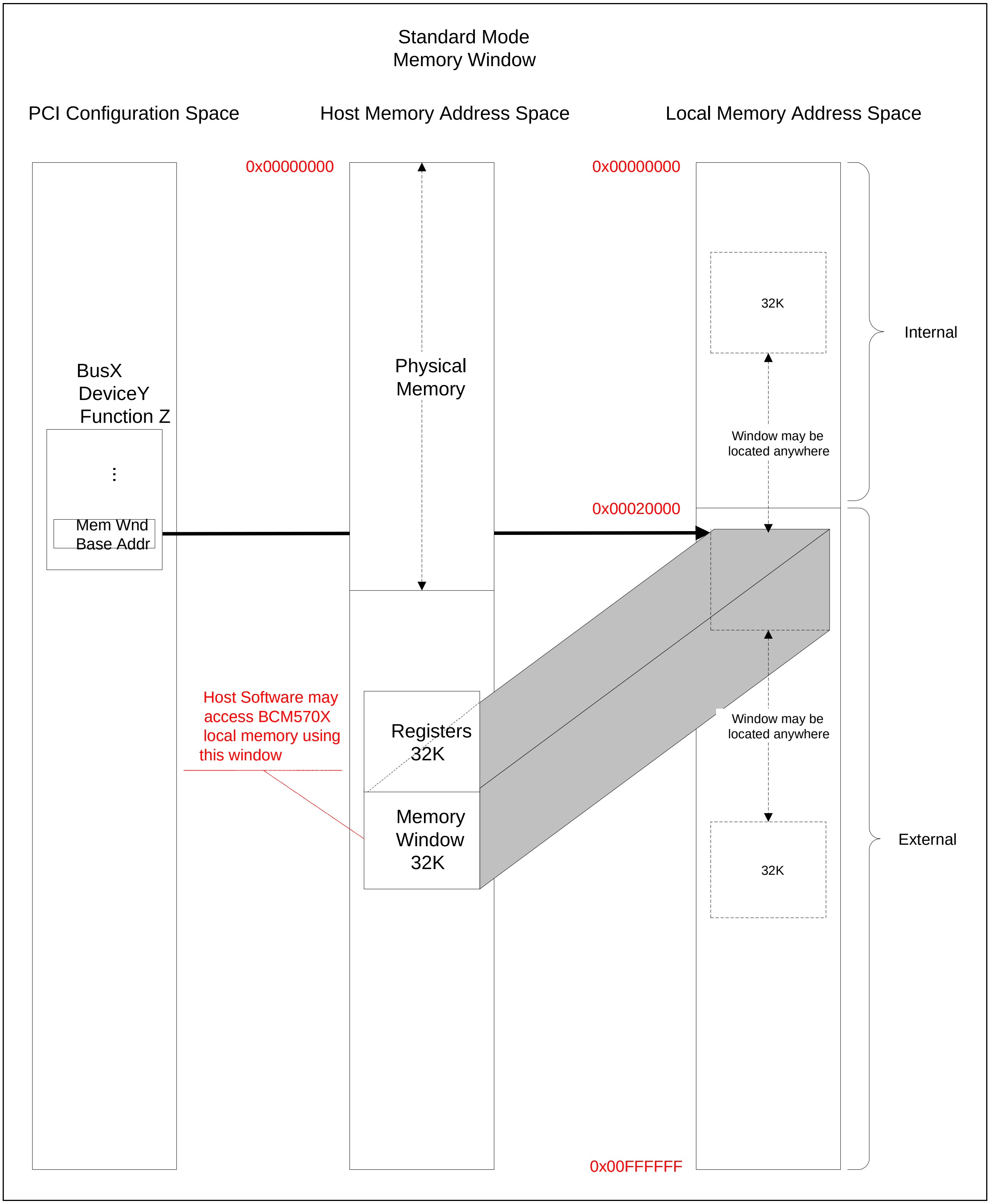


Figure 72: Standard Mode Memory Window

Flat Mode

Flat mode requires 32M of memory (see Figure 73). Flat mode is useful when host software wants all BCM57XX resources host memory mapped; diagnostic software is an example application. Mailboxes, Send Rings, Receive Rings, and Local Memory are directly mapped into the host memory space. Unlike standard mode, the BCM57XX family's local memory is addressable through host memory mapped I/O. Flat mode should not be used when there are limitations on the amount of memory mapped I/O available.

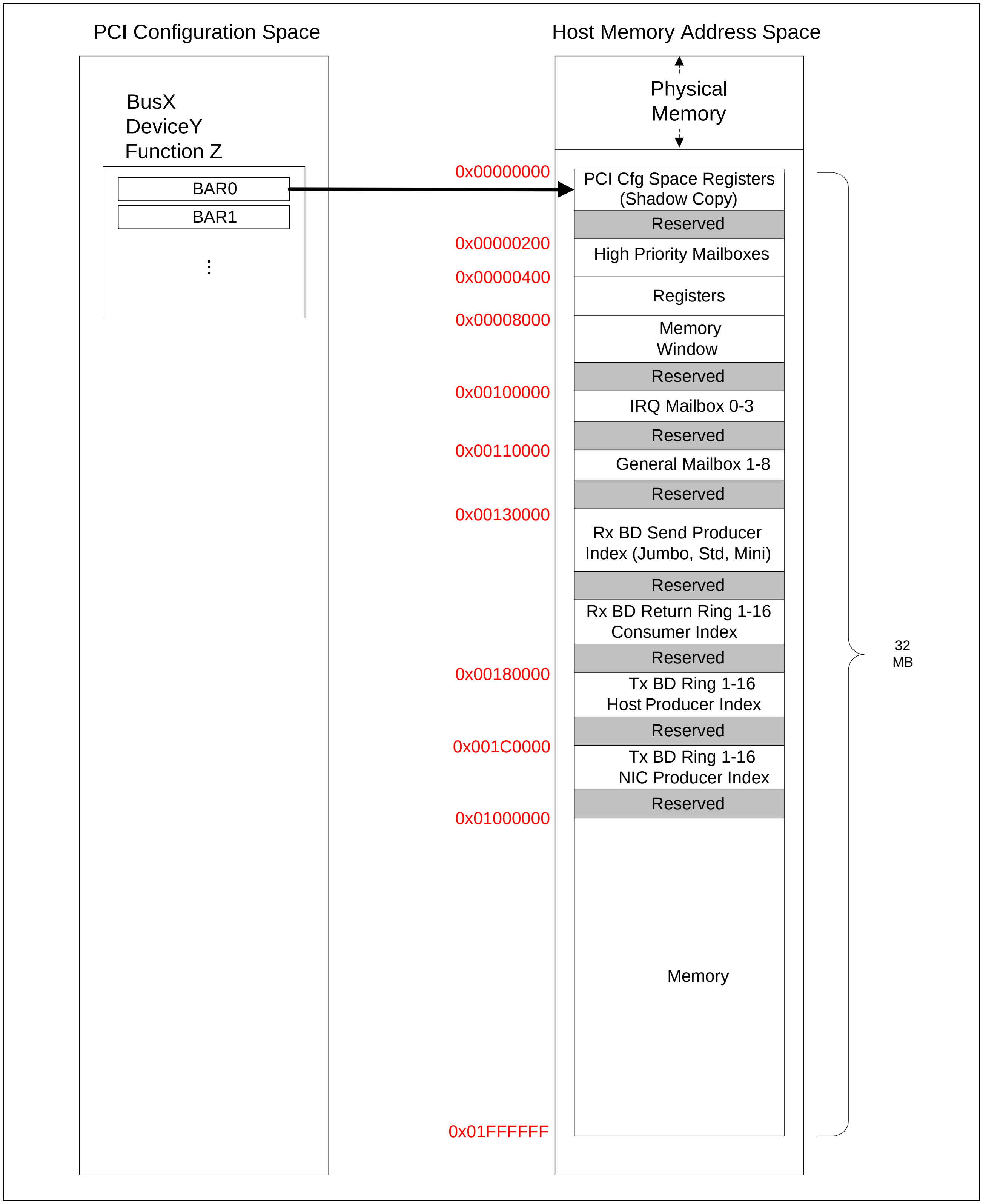


Figure 73: Flat Mode Memory Map

Table 89 shows the offsets which are relative to the PCI BAR—all host address decodes are BAR + OFFSET. Offsets 0x00-0xFF contain a shadow copy of the PCI configuration space registers. Host software may use this memory map to read/write to PCI configuration space registers. Address offsets 0x200-0x3FF are the high-priority mailboxes.

Table 89: PCI Address Map Flat View

Offset	*Name*	*Size*
0x00000000-0x000000FF	PCI Configuration space	256 bytes
0x00000200-0x000003FF	High Priority Mailboxes	512 bytes
0x00000400-0x00007FFF	BCM57XX registers	31 KB
0x00008000-0x0000FFFF	Memory Window	32 KB
0x00100000-0x00100007	Interrupt Mailbox 0	8 bytes
0x00104000-0x00104007	Interrupt Mailbox 1	8 bytes
0x00108000-0x00108007	Interrupt Mailbox 2	8 bytes
0x0010C000-0x0010C007	Interrupt Mailbox 3	8 bytes
0x00110000-0x00110007	General Mailbox 1	8 bytes
0x00114000-0x00114007	General Mailbox 2	8 bytes
0x00118000-0x00118007	General Mailbox 3	8 bytes
0x0011C000-0x0011C007	General Mailbox 4	8 bytes
0x00120000-0x00120007	General Mailbox 5	8 bytes
0x00124000-0x00124007	General Mailbox 6	8 bytes
0x00128000-0x00128007	General Mailbox 7	8 bytes
0x0012C000-0x0012C007	General Mailbox 8	8 bytes
0x00130000-0x00130007	Reserved	8 bytes
0x00134000-0x00134007	Receive BD Standard Producer Ring Producer Index	8 bytes
0x00138000-0x00138007	Receive BD Jumbo Producer Ring Producer Index	8 bytes
0x0013C000-0x0013C007	Receive BD Mini Producer Ring Producer Index	8 bytes
0x00140000-0x00140007	Receive BD Return Ring 1 Consumer Index	8 bytes
0x00144000-0x00144007	Receive BD Return Ring 2 Consumer Index	8 bytes
0x00148000-0x00148007	Receive BD Return Ring 3 Consumer Index	8 bytes
0x0014C000-0x0014C007	Receive BD Return Ring 4 Consumer Index	8 bytes
0x00150000-0x00150007	Receive BD Return Ring 5 Consumer Index	8 bytes
0x00154000-0x00154007	Receive BD Return Ring 6 Consumer Index	8 bytes
0x00158000-0x00158007	Receive BD Return Ring 7 Consumer Index	8 bytes
0x0015C000-0x0015C007	Receive BD Return Ring 8 Consumer Index	8 bytes
0x00160000-0x00160007	Receive BD Return Ring 9 Consumer Index	8 bytes
0x00164000-0x00164007	Receive BD Return Ring 10 Consumer Index	8 bytes
0x00168000-0x00168007	Receive BD Return Ring 11 Consumer Index	8 bytes
0x0016C000-0x0016C007	Receive BD Return Ring 12 Consumer Index	8 bytes
0x00170000-0x00170007	Receive BD Return Ring 13 Consumer Index	8 bytes
0x00174000-0x00174007	Receive BD Return Ring 14 Consumer Index	8 bytes
0x00178000-0x00178007	Receive BD Return Ring 15 Consumer Index	8 bytes
0x0017C000-0x0017C007	Receive BD Return Ring 16 Consumer Index	8 bytes
0x00180000-0x00180007	Send BD Ring 1 Host Producer Index	8 bytes
0x00184000-0x00184007	Send BD Ring 2 Host Producer Index	8 bytes

Table 89: PCI Address Map Flat View (Cont.)

Offset	Name	Size
0x00188000-0x00188007	Send BD Ring 3 Host Producer Index	8 bytes
0x0018C000-0x0018C007	Send BD Ring 4 Host Producer Index	8 bytes
0x00190000-0x00190007	Send BD Ring 5 Host Producer Index	8 bytes
0x00194000-0x00194007	Send BD Ring 6 Host Producer Index	8 bytes
0x00198000-0x00198007	Send BD Ring 7 Host Producer Index	8 bytes
0x0019C000-0x0019C007	Send BD Ring 8 Host Producer Index	8 bytes
0x001A0000-0x001A0007	Send BD Ring 9 Host Producer Index	8 bytes
0x001A4000-0x001A4007	Send BD Ring 10 Host Producer Index	8 bytes
0x001A8000-0x001A8007	Send BD Ring 11 Host Producer Index	8 bytes
0x001AC000-0x001AC007	Send BD Ring 12 Host Producer Index	8 bytes
0x001B0000-0x001B0007	Send BD Ring 13 Host Producer Index	8 bytes
0x001B4000-0x001B4007	Send BD Ring 14 Host Producer Index	8 bytes
0x001B8000-0x001B8007	Send BD Ring 15 Host Producer Index	8 bytes
0x001BC000-0x001BC007	Send BD Ring 16 Host Producer Index	8 bytes
0x001C0000-0x001C0007	Send BD Ring 1 NIC Producer Index	8 bytes
0x001C4000-0x001C4007	Send BD Ring 2 NIC Producer Index	8 bytes
0x001C8000-0x001C8007	Send BD Ring 3 NIC Producer Index	8 bytes
0x001CC000-0x001CC007	Send BD Ring 4 NIC Producer Index	8 bytes
0x001D0000-0x001D0007	Send BD Ring 5 NIC Producer Index	8 bytes
0x001D4000-0x001D4007	Send BD Ring 6 NIC Producer Index	8 bytes
0x001D8000-0x001D8007	Send BD Ring 7 NIC Producer Index	8 bytes
0x001DC000-0x001DC007	Send BD Ring 8 NIC Producer Index	8 bytes
0x001E0000-0x001E0007	Send BD Ring 9 NIC Producer Index	8 bytes
0x001E4000-0x001E4007	Send BD Ring 10 NIC Producer Index	8 bytes
0x001E8000-0x001E8007	Send BD Ring 11 NIC Producer Index	8 bytes
0x001EC000-0x001EC007	Send BD Ring 12 NIC Producer Index	8 bytes
0x001F0000-0x001F0007	Send BD Ring 13 NIC Producer Index	8 bytes
0x001F4000-0x001F4007	Send BD Ring 14 NIC Producer Index	8 bytes
0x001F8000-0x001F8007	Send BD Ring 15 NIC Producer Index	8 bytes
0x001FC000-0x001FC007	Send BD Ring 16 NIC Producer Index	8 bytes
0x01000000-0x01FFFFFF	Memory	16 MB

The proceeding host memory ranges are relative to the BCM57XX PCI BAR—the address decodes are BAR + offset. Offsets 0x00-0xFF contain a shadow copy of the PCI configuration space registers. Host software may use this memory map to read/write to PCI configuration space registers. Address offsets 0x200-0x3FF are high priority mailboxes.

There are several similarities to standard mode. First, BCM57XX registers are memory-mapped into the host offsets 0x400-0x7FFF. See "Register Definitions" on page 298 for complete register and bit definitions. Second, the 32K memory window is available in flat mode. However, host software should just use the direct memory map, at BAR + offsets 0x01000000-0x01FFFFFF. The direct memory map requires no setup (PCI Memory Base Address register), and is always available in flat mode. See "Standard Mode" on page 192 for the discussion of the memory window in the standard mode.

Figure 74 on page 200 shows the Flat Mode Memory map.

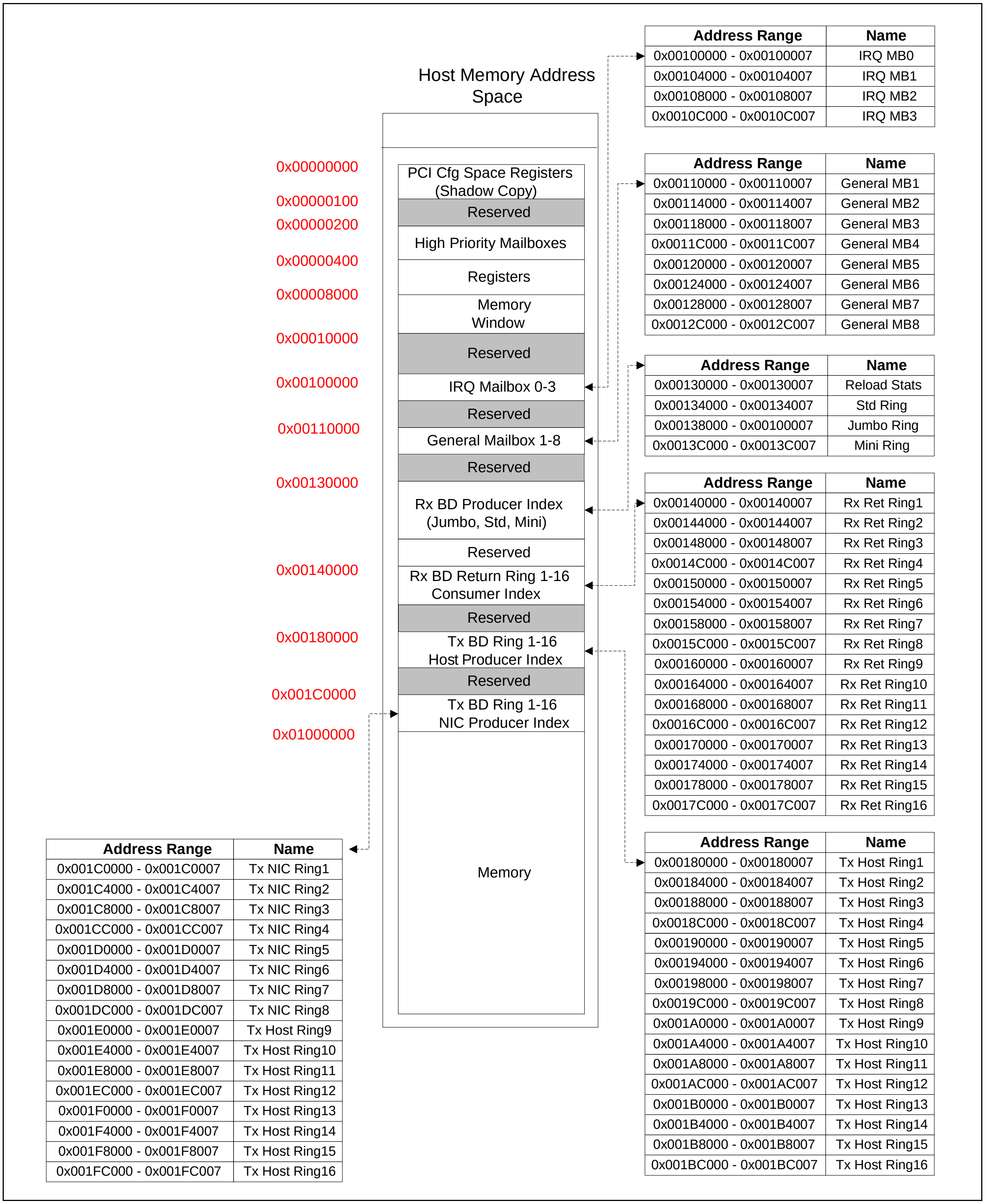
The interrupt mailboxes are accessible from BAR + offsets 0x00100000 to 0x0010C007. Host software may use this memory map to access the interrupt mailboxes located in BCM57XX family's register block (0x200-0x21F). The Receive Producer (mini, standard, jumbo) consumer index mailboxes are available at offsets 0x00130000-0x0013C0007. Receive return ring mailboxes are shadowed at BAR + offsets 0x00140000 to 0x0017C007. The BCM57XX family receive return rings are located in register block range 0x280-0x2FF. Host-based send rings are located at BAR + offsets 0x001800000–0x001BC007. These sent mailboxes are mapped from the BCM57XX register block range 0x300-0x37F. Finally, the NIC-based send rings start at offset 0x001C0000 and end at offset 0x001FC007. Again, the NIC-based send rings are mapped from the BCM57XX register space addresses 0x380–0x3FF.

The address ranges 0x01000000-0x01FFFFFF are aliases for the BCM57XX device local memory. This includes both internal and external memory—the entire device memory region. When host software reads/writes from address ranges starting at BAR + 0x01000000, the address is normalized/mapped to a device local address starting at 0x00000000. For example, the host address 0x010000FF is translated to device local address 0x000000FF.

Figure 75 on page 201 shows the two mechanisms host software may use to access BCM57XX local memory (flat mode configuration).

- First, host software can use the memory window.
- Second, host software may access the memory-mapped range 0x01000000 to 0x01FFFFFF.

The second technique is preferable since flat mode has made 32M of host memory-mapped space available. Software does not need to use the memory window since the BCM57XX family's entire memory region is memory-mapped.



Address Range	Name
0x00100000 - 0x00100007	IRQ MB0
0x00104000 - 0x00104007	IRQ MB1
0x00108000 - 0x00108007	IRQ MB2
0x0010C000 - 0x0010C007	IRQ MB3

Address Range	Name
0x00110000 - 0x00110007	General MB1
0x00114000 - 0x00114007	General MB2
0x00118000 - 0x00118007	General MB3
0x0011C000 - 0x0011C007	General MB4
0x00120000 - 0x00120007	General MB5
0x00124000 - 0x00124007	General MB6
0x00128000 - 0x00128007	General MB7
0x0012C000 - 0x0012C007	General MB8

Address Range	Name
0x00130000 - 0x00130007	Reload Stats
0x00134000 - 0x00134007	Std Ring
0x00138000 - 0x00100007	Jumbo Ring
0x0013C000 - 0x0013C007	Mini Ring

Address Range	Name
0x00140000 - 0x00140007	Rx Ret Ring1
0x00144000 - 0x00144007	Rx Ret Ring2
0x00148000 - 0x00148007	Rx Ret Ring3
0x0014C000 - 0x0014C007	Rx Ret Ring4
0x00150000 - 0x00150007	Rx Ret Ring5
0x00154000 - 0x00154007	Rx Ret Ring6
0x00158000 - 0x00158007	Rx Ret Ring7
0x0015C000 - 0x0015C007	Rx Ret Ring8
0x00160000 - 0x00160007	Rx Ret Ring9
0x00164000 - 0x00164007	Rx Ret Ring10
0x00168000 - 0x00168007	Rx Ret Ring11
0x0016C000 - 0x0016C007	Rx Ret Ring12
0x00170000 - 0x00170007	Rx Ret Ring13
0x00174000 - 0x00174007	Rx Ret Ring14
0x00178000 - 0x00178007	Rx Ret Ring15
0x0017C000 - 0x0017C007	Rx Ret Ring16

Address Range	Name
0x00180000 - 0x00180007	Tx Host Ring1
0x00184000 - 0x00184007	Tx Host Ring2
0x00188000 - 0x00188007	Tx Host Ring3
0x0018C000 - 0x0018C007	Tx Host Ring4
0x00190000 - 0x00190007	Tx Host Ring5
0x00194000 - 0x00194007	Tx Host Ring6
0x00198000 - 0x00198007	Tx Host Ring7
0x0019C000 - 0x0019C007	Tx Host Ring8
0x001A0000 - 0x001A0007	Tx Host Ring9
0x001A4000 - 0x001A4007	Tx Host Ring10
0x001A8000 - 0x001A8007	Tx Host Ring11
0x001AC000 - 0x001AC007	Tx Host Ring12
0x001B0000 - 0x001B0007	Tx Host Ring13
0x001B4000 - 0x001B4007	Tx Host Ring14
0x001B8000 - 0x001B8007	Tx Host Ring15
0x001BC000 - 0x001BC007	Tx Host Ring16

Address Range	Name
0x001C0000 - 0x001C0007	Tx NIC Ring1
0x001C4000 - 0x001C4007	Tx NIC Ring2
0x001C8000 - 0x001C8007	Tx NIC Ring3
0x001CC000 - 0x001CC007	Tx NIC Ring4
0x001D0000 - 0x001D0007	Tx NIC Ring5
0x001D4000 - 0x001D4007	Tx NIC Ring6
0x001D8000 - 0x001D8007	Tx NIC Ring7
0x001DC000 - 0x001DC007	Tx NIC Ring8
0x001E0000 - 0x001E0007	Tx Host Ring9
0x001E4000 - 0x001E4007	Tx Host Ring10
0x001E8000 - 0x001E8007	Tx Host Ring11
0x001EC000 - 0x001EC007	Tx Host Ring12
0x001F0000 - 0x001F0007	Tx Host Ring13
0x001F4000 - 0x001F4007	Tx Host Ring14
0x001F8000 - 0x001F8007	Tx Host Ring15
0x001FC000 - 0x001FC007	Tx Host Ring16

Figure 74: Flat Mode Memory Map

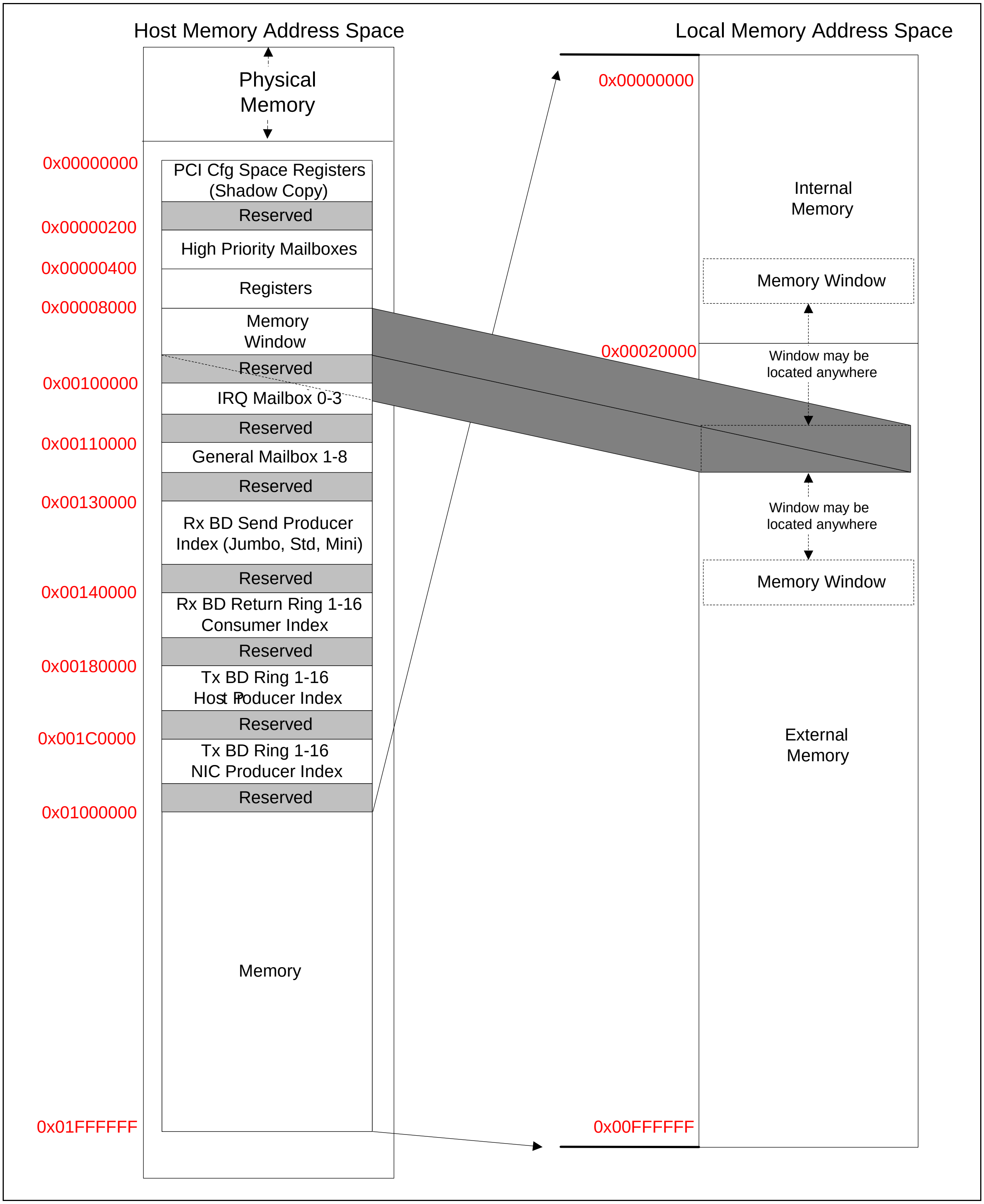


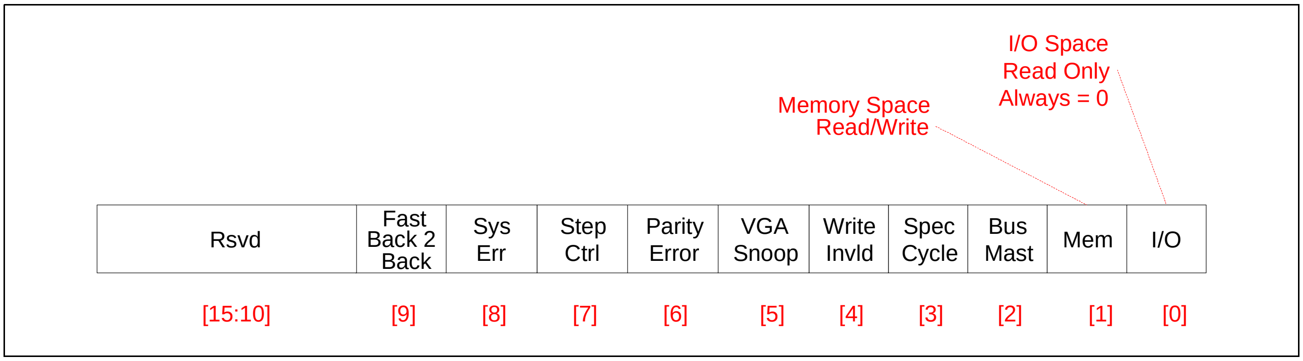
Figure 75: Techniques for Accessing BCM57XX Local Memory

Memory Mapped I/O Registers

The following BCM57XX registers are used in the mode configuration of the PCI memory-mapped I/O.

PCI Command Register

The PCI_Command register is 16 bits wide (see Figure 76). The BCM57XX family does not have I/O mapped I/O. The I/O_Space bit is de-asserted by hardware. The BCM57XX family does support Memory_Mapped_Memory and hardware will assert the Memory_Space bit. Both these bits are read-only and are usually read by the PnP BIOS/OS. The BIOS/OS examines these bits to assign non-conflicting resources to PCI devices.



Rsvd	Fast Back 2 Back	Sys Err	Step Ctrl	Parity Error	VGA Snoop	Write Invld	Spec Cycle	Bus Mast	Mem	I/O
[15:10]	[9]	[8]	[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]

Figure 76: PCI Command Register

PCI State Register

The PCI_State register is 32 bits wide. Operating mode is set with the Flat_View bit in the PCI_State register. When the Flat_View bit is asserted, the BCM57XX family decodes a 32M of block host memory. When the Flat_View bit is de-asserted, the BCM57XX family decodes a 64K block of host memory.

PCI Base Address Register

The PCI_Base_Address register (BAR) specifies the location of a BCM57XX memory mapped I/O block. The BCM57XX mode configuration (Flat vs. Standard) affects how the BAR is setup (see Figure 77).

- Bits 4-31 in the PCI_Base_Address register are selectively programmable based on the amount of host memory requested. The PnP BIOS/OS will use an algorithm to test the BAR bits and determine the amount of physical memory requested.
- The Memory_Space_Indicator bit designates whether the BAR is memory or I/O mapped. The BCM57XX hard codes the Memory_Space_Indicator bit to zero (de-asserted).
- The Location/Type bits specify locations in host memory space where a device can decode physical addresses. The BCM57XX memory mapped I/O range may be placed anywhere in 64-bit address space (Type = 10).
- The BCM57XX family de-asserts the Prefetchable bit to indicate that the memory range should not be cached.

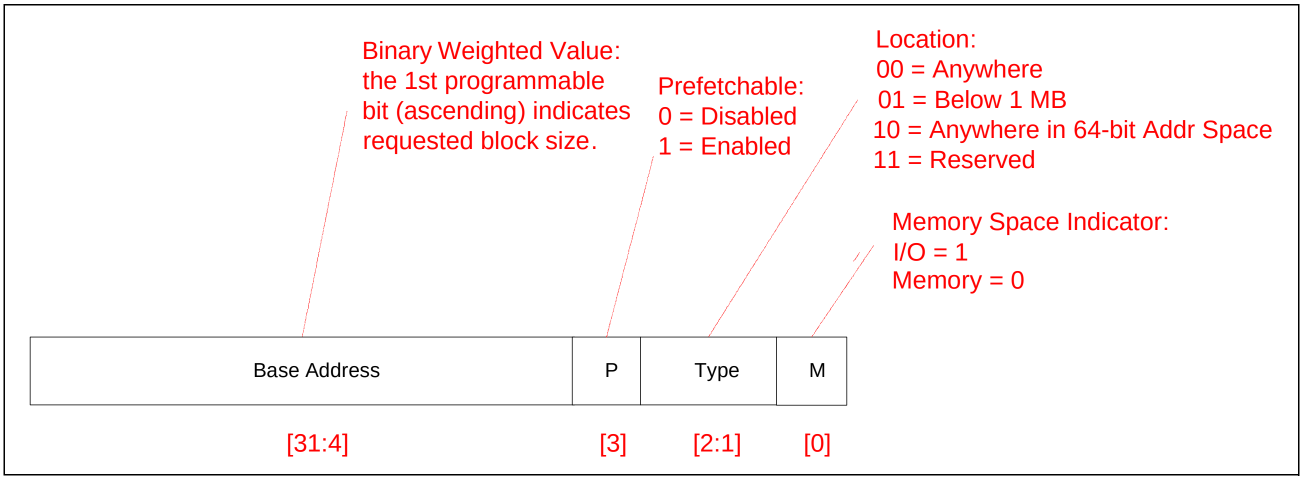


Figure 77: PCI Base Address Register

The BCM57XX 64K memory mapped I/O block is determined by the first programmable bit in the BAR. When the MAC is configured in standard mode, the mask 0xFFFF0000 identifies the BAR bits, which are programmable. Bit 16 is the first bit encountered in the scan upward, which is programmable; bits 0-3 are ignored. Host software will read zero values from bits 4-16. Figure 78 shows the BAR register and the bits returned to the OS/BIOS during resource allocation.

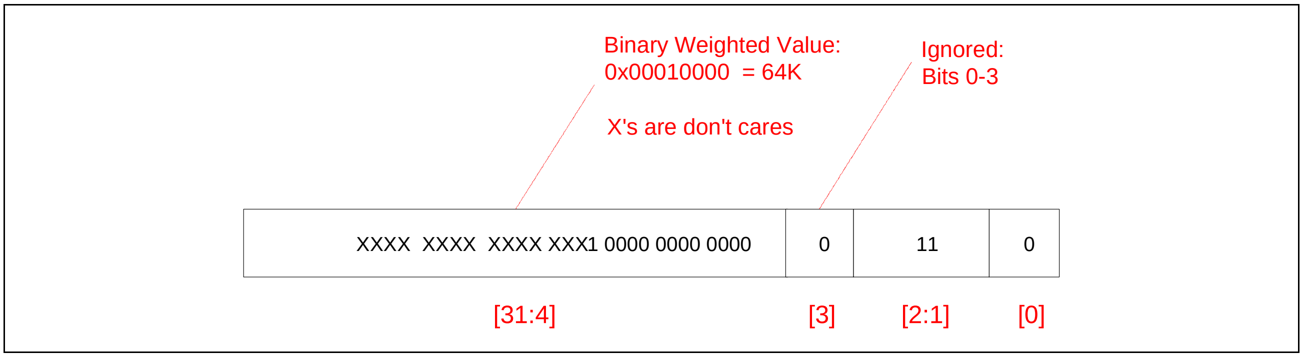


Figure 78: PCI Base Address Register Bits Read in Standard Mode

When the BCM57XX MAC is programmed in flat mode, a 32M region of memory mapped I/O needs to be made available. The PnP BIOS/OS will probe the BAR, and scan upwards looking for the first programmable bit. Again, bits 0-3 are ignored. The mask 0xE000000 identifies the BAR bits, which are programmable. Bit 25 is the first bit encountered in the scan upward, which is programmable. Host software will read zero values from bits 4-24. Figure 79 shows the BAR register and the bits returned to the OS/BIOS during resource allocation.

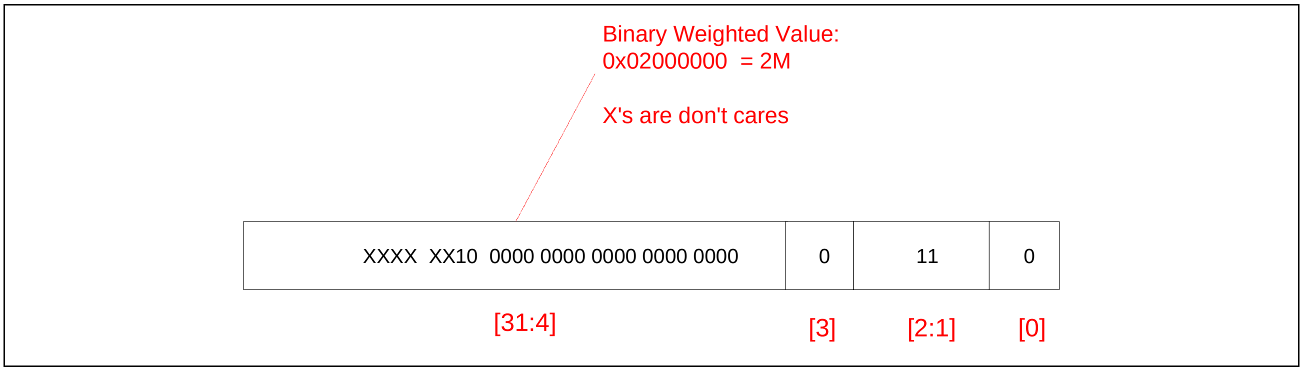


Figure 79: PCI Base Address Register Bits Read in Flat Mode

Register Quick Cross Reference

BCM57XX Family

The BCM57XX PCI registers are listed in Table 90.

Table 90: BCM57XX PCI Registers

Register	*Bit*	*Cross Reference*
PCI Command	Memory_Space	See "Command Register (Offset 0x04)" on page 302.
PCI Command	IO_Space	See "Command Register (Offset 0x04)" on page 302.
PCI State	Flat_View	See "PCI State Register (Offset 0x70)" on page 332.
PCI Base Address 1	All	See "Base Address Register 1/2 Register (Offset 0x10-0x17)" on page 306.
PCI Base Address 2	All	See "Base Address Register 1/2 Register (Offset 0x10-0x17)" on page 306.

PSEUDOCODE

- Variable BCM57XXMemAddr represents a local memory address.
- Variable BCM57XXMemBase represents a 32K aligned local memory address.
- Variable BCM57XXMemOffset represents a byte offset.
- PCI_CFG_WRITE(address, *value*) is a routine to write the device's PCI configuration space register.
- PCI_CFG_READ(address) is a routine to return the contents of the device's PCI configuration space register.

Memory Window Read in Standard Mode

```
BCM57XXMemBase = BCM57XXMemAddr & 0xFFFF0000
BCM57XXMemOffset = BCM57XXMemAddr & 0x0000FFFF
PCI_CFG_WRITE(Memory_Window_Base_Address, BCM57XXXMemBase)
Value = *(PCI_BAR + 0x8000 + BCM57XXMemOffset)
```

Memory Window Write in Standard Mode

```
BCM57XXMemBase = BCM57XXMemAddr & 0xFFFF0000
BCM57XXMemOffset = BCM57XXMemAddr & 0x0000FFFF
PCI_CFG_WRITE(Memory_Window_Base_Address, BCM57XXXMemBase)
*(PCI_BAR + 0x8000 + BCM57XXMemOffset) = Value
```

Register Read in Standard Mode

```
Value = *(PCI_BAR + 0x400 + BCM57XX_REGISTER_OFFSET)
```

Register Write in Standard Mode

```
*(PCI_BAR + 0x400 + BCM57XX_REGISTER_OFFSET) = Value
```

Memory Read in Flat Mode[1]

```
Value = *(PCI_BAR + 0x01000000 + BCM57XXMemAddr)
```

Memory Write in Flat Mode[1]

```
*(PCI_BAR + 0x01000000 + BCM57XXMemAddr) = Value
```

Register Read in Flat Mode

```
Value = *(PCI_BAR + 0x400 + BCM57XX_REGISTER_OFFSET)
```

Register Write in Flat Mode

```
*(PCI_BAR + 0x400 + BCM57XX_REGISTER_OFFSET) = Value
```

Memory Read Using Indirect Mode

```
PCI_CFG_WRITE(Memory_Window_Base_Address, BCM57XXXMemAddr)
Value = PCI_CFG_READ(Memory_Window_Data)
```

Memory Write Using Indirect Mode

```
PCI_CFG_WRITE(Memory_Window_Base_Address, BCM57XXXMemAddr)
PCI_CFG_WRITE(Memory_Window_Data, Value)
```

[1] The scratchpad and ROM regions must be accessed using indirect register pairs.

Register Read Using Indirect Mode

```
PCI_CFG_WRITE(Register_Base_Address, BCM57XXXRegAddr)
Value = PCI_CFG_READ(Register_Data_Register)
```

Register Write Using Indirect Mode

```
PCI_CFG_WRITE(Register_Base_Address, BCM57XXXRegAddr)
PCI_CFG_WRITE(Register_Data_Register, Value)
```

BUS INTERFACE

DESCRIPTION

The read/write DMA engines both drive the PCI interface. Normally, each DMA engine alternates bursts to the PCI bus, and both interfaces may have outstanding transactions on the PCI bus. The BCM57XX architecture identifies two channels—a read DMA channel and a write DMA channel. Each channel corresponds to the appropriate DMA engine (see Figure 80). The configuration of the DMA engines and the PCI interface is discussed in this section.

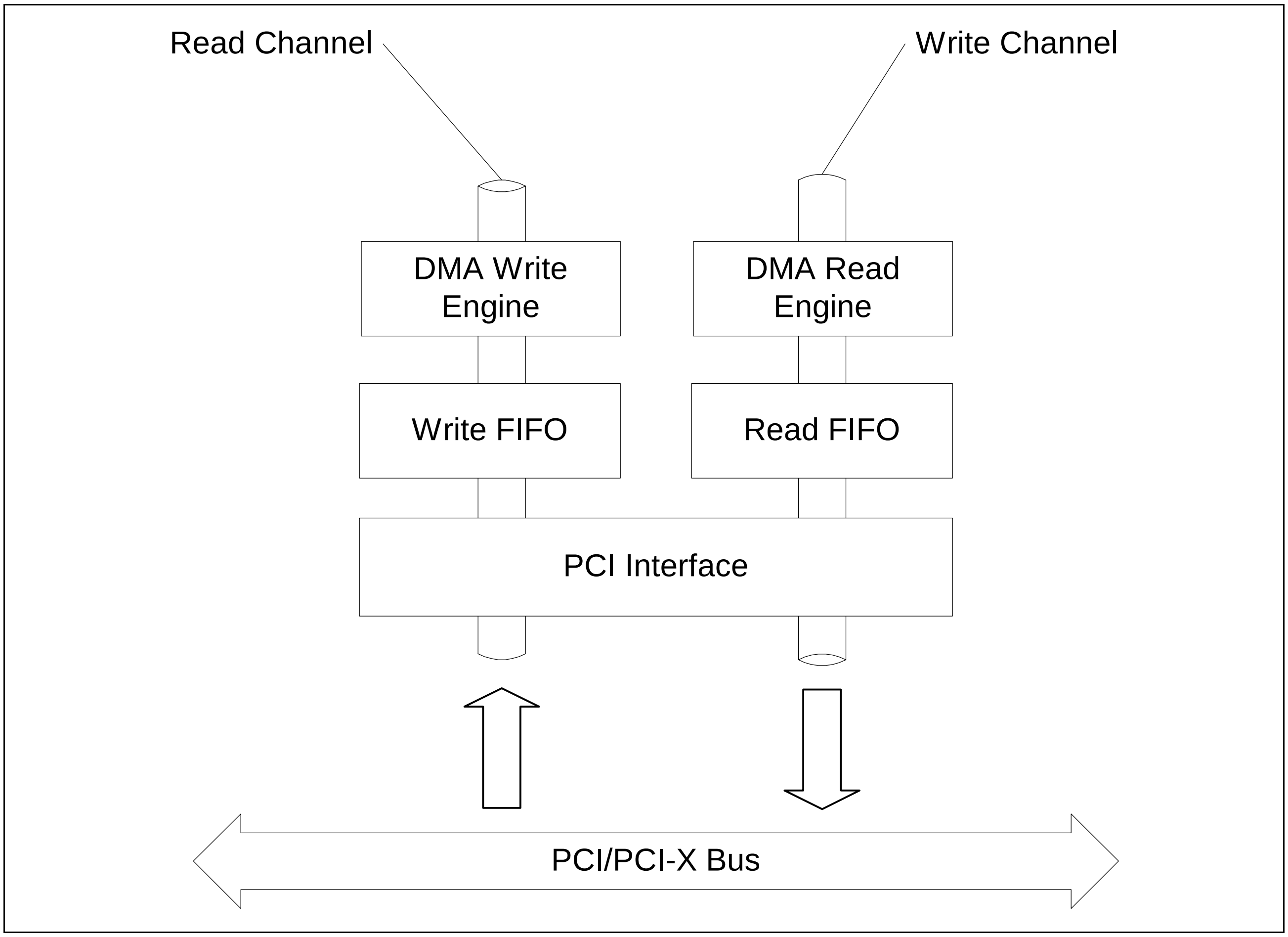


Figure 80: Read and Write Channels of DMA Engine

The following architectural components are involved in the configuration of the PCI/DMA interface:

- DMA Read Engine
- DMA Write Engine
- DMA Read FIFO
- DMA Write FIFO
- PCI Interface and Connector
- PCI State register
- DMA Read/Write register

OPERATIONAL CHARACTERISTICS

Bus Mode Detection

The Conventional_PCI bit is for status and is located in the PCI State register (see "PCI State Register (Offset 0x70)" on page 332). This bit reflects the operational mode of the PCI/PCI-X interface. Software may read this bit to determine the operational state of the BCM57XX PCI interface. When this bit is cleared, the BCM57XX family supports PCI-X operation. The MAC uses a standard PCI protocol when this bit is asserted. Host software should not set this bit and may read this bit to determine current PCI/PCI-X operational mode.

A PCI bus may be either 32 or 64 bits wide. The BCM57XX family will detect bus width and configure itself to use the widest bus topology available. The BCM57XX bus width is initialized after device reset based on the sampling of $\overline{\text{REQ64}}$. The 32_Bit_Bus_Mode bit is read only and is located in the PCI State register. This bit reflects the bus width of the BCM57XX PCI/PCI-X interface.

If a 32-bit bus mode is detected, the $\overline{\text{REQ64}}$ signal is not asserted by the PCI interface. When the $\overline{\text{REQ64}}$ is asserted, 32_Bit_PCI_Bus bit is clear. Additionally, the PCI interface will not assert $\overline{\text{ACK64}}$ as a transaction target response. The BCM57XX family will also drive the $\overline{\text{ACK64}}$ signal, when a target response is required. Figure 81 shows a timing diagram where the 32_Bit_PCI_Bus bit is shown to affect the $\overline{\text{REQ64}}$ and $\overline{\text{ACK64}}$ signals.

The BCM57XX family supports 33/66 MHz PCI and 66/133 MHz PCI-X bus speeds. The 33/66PCI_66/133PCIX_Mode bit is read only and is also located in the PCI State register. This bit reflects both PCI and PCI-X bus speeds. A deasserted bit indicates the PCI interface operates at 33 MHz or the PCI-X interface operates at 66 MHz. An asserted bit indicates the PCI interface operates at 66 MHz or the PCI-X interface operates between 66-133 MHz. Host software may read this bit to determine the bus frequency sampled by the BCM57XX bus interface.

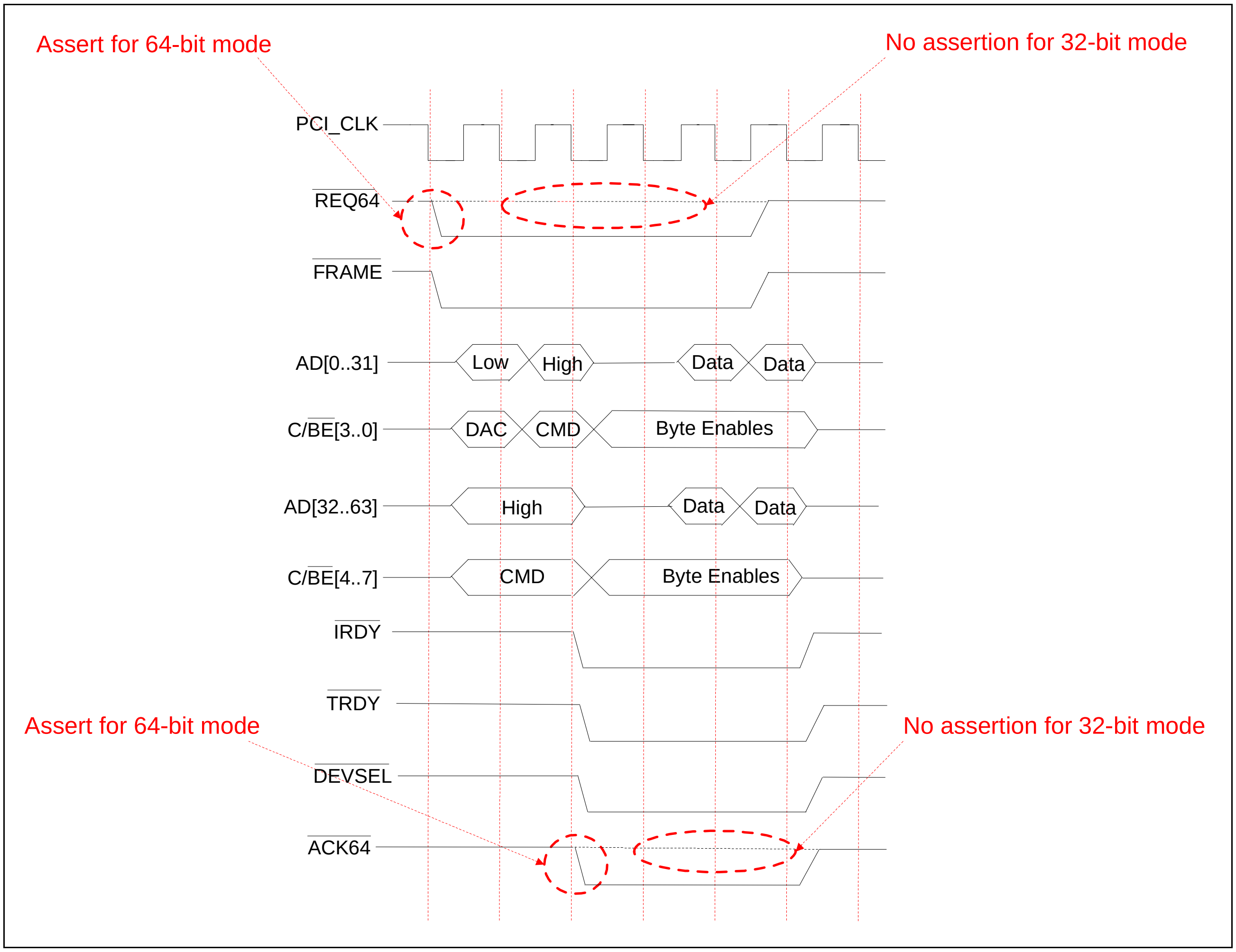


Figure 81: 32_Bit_PCI_Bus Bit Affects the $\overline{\text{REQ64}}$ and $\overline{\text{ACK64}}$ Signals

PCI Command Usage

Host software may override the default command encoding for PCI read and write transactions. The BCM57XX family drives a command word onto the C/BE lines during the address phase of a transaction (see Figure 82).

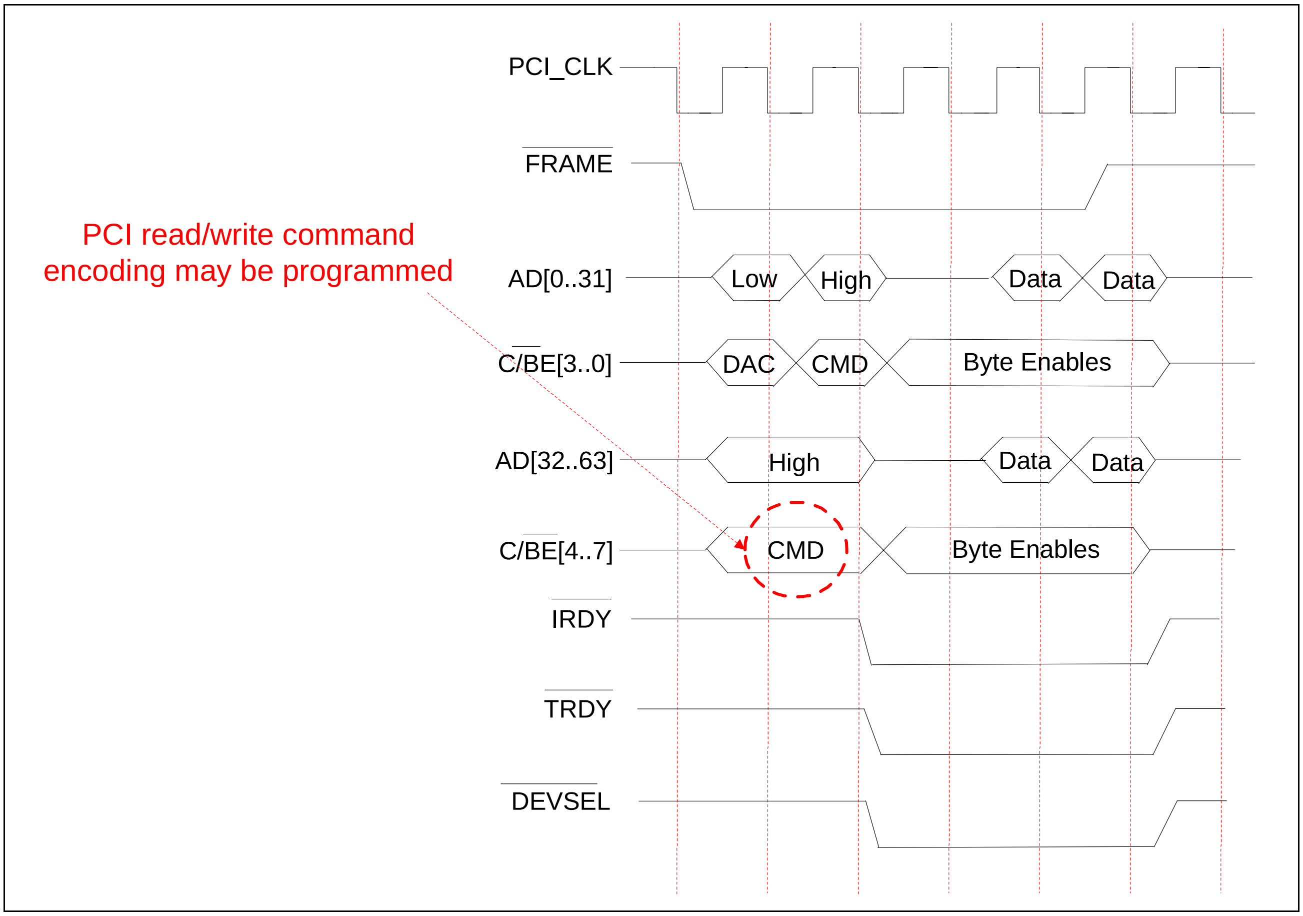


Figure 82: Host Software Can Override Default PCI Command Encoding

The Default_PCI_Write_Command and Default_PCI_Read_Command bits in the DMA Read/Write Control register (see "DMA Read/Write Control Register (Offset 0x6C)" on page 327) are read/write. The MAC defaults the Default_PCI_Write_Command bits to 0x00 on reset. The value 0x7 is the PCI v2.2 specified encoding for a PCI Write. The Default_PCI_Read_Command bits default to 0x00 at device reset. The value 0x6 is the PCI v2.2 specified encoding for a PCI read.

Note: For the BCM57XX family, it is recommended that host software use the following settings listed in Table 91 for device initialization. Any change to these register bits may seriously affect PCI bus transactions; therefore, it is not recommended that host software use these values.

Table 91: Recommended Setting for PCI Command Encoding

Register	***Bit Field Name***	***Recommended Value***
DMA Read/Write Control	Default_PCI_Write_Command	0x07
DMA Read/Write Control	Default_PCI_Read_Command	0x06

The BCM57XX family may align PCI transactions using the command byte enable signals on the PCI Bus. The Assert_All_BE_On_DMA_Write bit is also located in the DMA Read/Write Control register and this bit is read/write. Software may set this bit to force all C/$\overline{BE}$ lines to be asserted during a PCI Write transaction. For example, the PCI interface on the BCM57XX family may only need to write one byte to address 0x0003. When the BCM57XX family is operating in 32-bit mode, the PCI interface will assert C/$\overline{BE}$[3] and drive valid data on AD[24..31]. The master still needs to drive all 32-bit address lines (and parity) even though it only asserts one BE. When this bet is set, the master will assert all C/$\overline{BE}$[0..3] lines (see Figure 83). The address block 0x0000 to 0x0003 will be written. Enabling the Assert_All_Bes_On_DMA bit requires the BCM57XX family to drive valid data on AD[0..31].

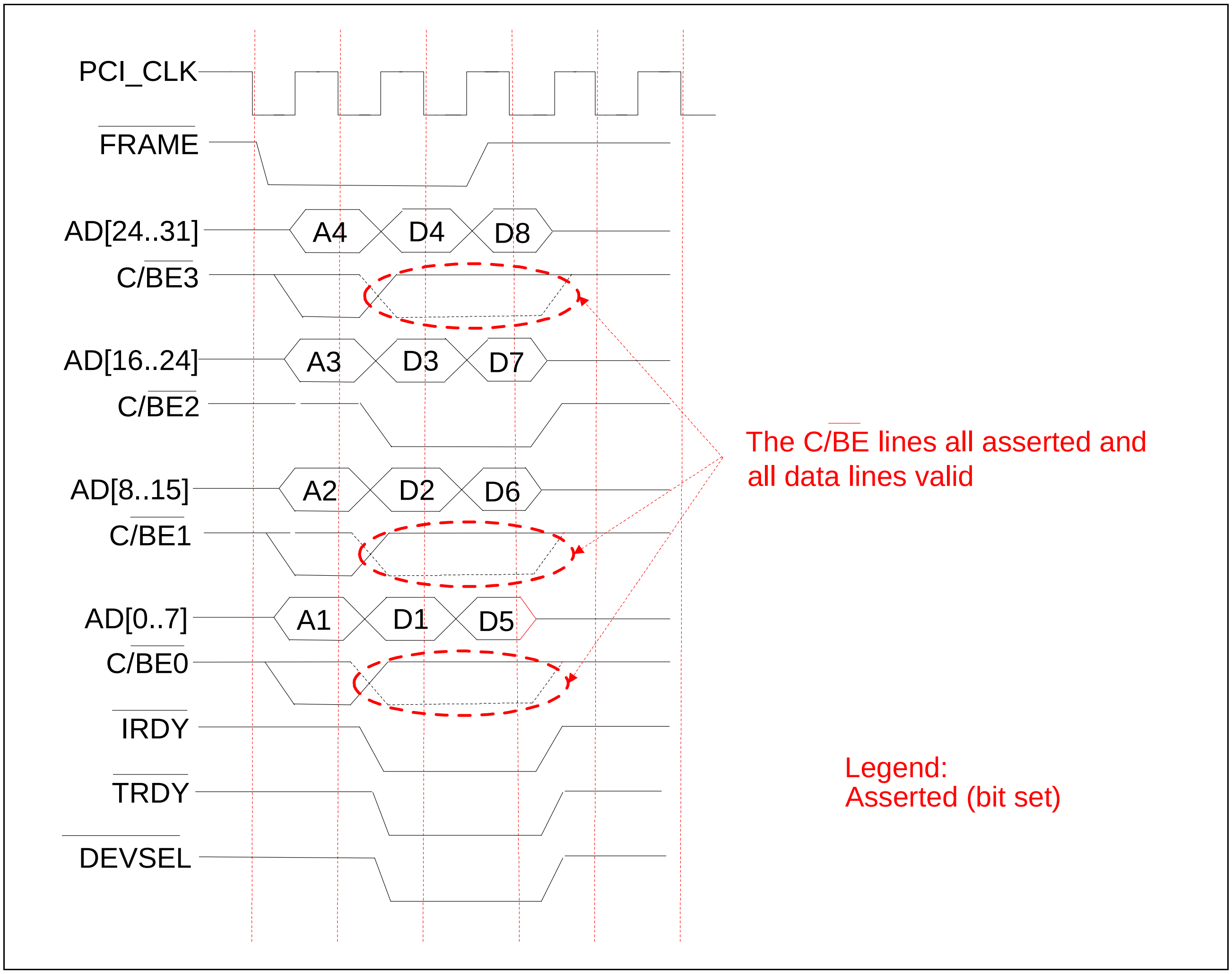


Figure 83: All C/BE[0..3] Lines Asserted and All Data Lines Valid

The Memory Write and Invalidate (MWI) PCI command requires the master to terminate all write transactions on cache line boundary. For example, the Intel Pentium III architecture has a cache line size of 32 bytes. The DMA write engine will terminate all Memory Write and Invalidate transactions on a 32-byte boundary; the PCI v2.2 specification states that MWI commands must terminate on a cache line. Host software can enable MWI by setting the Memory_Write_And_Invalidate bit in the PCI configuration space Command register (see "Command Register (Offset 0x04)" on page 302). This bit defaults to disabled after reset. The reader is strongly encouraged to reference Broadcom errata notes for the device being used.

The Memory Read (MR) command is used when a transaction data phase is two cycles or less. The MR command does not allow a host PCI bridge to prefetch data. The PCI interface may use the Memory Read Line (MRL) command when a transaction data phase is between two cycles and one cache line long—the host PCI bridge may prefetch up to a cache line worth of data. A transaction data phase that exceeds one cache line uses the Memory Read Multiple (MRM) PCI command. The host PCI bridge will prefetch data for a MRM command while the master asserts FRAME. The MRM command provides the host PCI bridge the performance gain of prefetching data beyond a cache line. Both the MRL and MRM commands do not have to read an entire cache line, and the master may terminate the transaction before the entire cache line is read.

Host software may configure the PCI interface to use the Memory Read Multiple in place of the Memory Read Line command. The Use_MemRdMult_Command bit is read/write and located in the DMA Read/Write Control register (see "DMA Read/Write Control Register (Offset 0x6C)" on page 327). This bit defaults to zero after device reset. Software can use this bit to disable the PCI Memory Read Multiple command on the BCM57XX family.

Read/Write DMA Engines

Software must enable the bus master DMA bit for the BCM57XX family. The BCM57XX family is a bus-mastering device and the PCI interface requires that the Bus_Master enable bit be set by either the BIOS or host device driver. A bus master drives the transaction boundary signal $\overline{\text{FRAME}}$ and is responsible for sourcing the address phase of a transaction. The bus master is the PCI transaction initiator. A PCI target will claim the transaction driven by the bus master. The Bus_Master enable bit is located in the PCI configuration space Command register (see "Command Register (Offset 0x04)" on page 302) and this bit is read/write. The bit defaults to cleared/disabled after device reset.

A PCI bus master may terminate a transaction under two conditions: completion and timeout. A completion occurs when all the necessary data has been read/written to the PCI target. A timeout occurs when the Master Latency Timer (MLT) expires. The memory write and invalidate transaction may not be terminated immediately should the MLT expire. The memory write and invalidate must complete on a cache line boundary. The memory write and invalidate transaction is an exception to a MLT timeout.

A PCI master must normally terminate a transaction after its grant line ($\overline{\text{GNT}}$) and the MLT expire. The BCM57XX DMA read/write engines may be programmed to end transactions on a boundary address regardless of termination conditions. Host software may align addresses for transaction termination for many reasons:

- The master may maximize the prefetch capability of a target device. Only memory read line and memory read multiple PCI transactions allow target prefetch. The PCI target may prefetch data into a FIFO, so the data may be burst efficiently on the PCI bus. Otherwise, a target device will insert wait states into the transaction.
- Host software may need to program these registers should a transaction target be a streaming device. Once data is prefetched into a read/write FIFO, streaming data cannot be recovered. Host software may need to write a certain number of bytes per DMA without exception.

Both the DMA_Read_Address_Boundary and DMA_Write_Address_Boundary bit fields are located in the DMA Read Write register (see "DMA Read/Write Control Register (Offset 0x6C)" on page 327) and these bit fields are read/write. The bit fields are 3-bit encoded and default to 0x000 after power on reset. The bit field encodings have different meanings based on PCI vs PCI-X bus context. Host software should be careful to check the detected bus configuration before programming these bit fields. Otherwise, DMA read/write transaction termination may not align on a desired boundary. By setting read/write DMA boundary registers, the device will always terminate whenever it hits that boundary regardless of whether it still has more data it wants to DMA, and regardless of whether its MLT has expired or whether it still has $\overline{\text{GNT}}$.

In general, Broadcom does not recommend that software set these registers because they will usually result in shorter bursts that will hurt overall performance.

The BCM57XX architecture defines two DMA channels—a read and write channel. Each DMA engine must share a common PCI interface and arbitrate for bus bandwidth accordingly. Host software may prevent both the Read and Write DMA engines from driving simultaneous transactions on the PCI bus. A deferred PCI transaction or a split PCI-X transaction is a situation where both DMA engines may drive simultaneous bus transactions. Host software can serialize the DMA engines so each engine must wait for the previous engine to complete deferred or split transactions. Once a transaction completes, the waiting DMA engine can initiate its burst transaction. The One_DMA_At_Once bit is read/write and is also located in the DMA Read Write Control register. This register defaults to a value of zero on reset; both engines may alternate bursts.

The read and write engines may both access the PCI interface. The PCI interface normally alternates burst traffic between both engines. Once the burst transaction is terminated, the other DMA engine is granted access to the PCI interface. Each engine's transaction may be either master or target terminated. A PCI target may delay a PCI transaction by terminating with a retry; the master may consider the transaction outstanding. A PCI-X transaction may be terminated with a split completion and the target will burst data back to the master later (see "PCI-X Host Bus Interface (Applicable to BCM5700, BCM5701, BCM5703C, BCM5703S, BCM5704C, and BCM5704S)" on page 83). Both scenarios allow the other DMA engine to initiate a new transaction.

The Minimum_DMA bit field is read/write and is also located in the DMA Read/Write register. This register's unit of measure is scaled to the PCI bus width. A 32-bit bus associates PCI WORD (32-bit) units to this bit field. A 64-bit bus associates PCI DWORD (64-bit) units to this bit field. Software should check the 32_Bit_Bus_Mode bit in the PCI_State register (see "PCI State Register (Offset 0x70)" on page 332) before programming the Minimum_DMA field.

The Minimum_DMA bit field configures the minimum number of PCI words burst before the PCI/PCI-X interface is relinquished to the other DMA channel. This configuration allows one DMA engine to complete a burst before preemption by the other DMA engine. The preemption is for the physical PCI interface since each DMA channel is independently governed. Only one channel can drive the physical PCI interface at any moment. Larger DMA bursts generally utilize PCI/PCI-X bandwidth more efficiently; however, software increases DMA read/write latency (not PCI transaction latency) by setting a larger value in the Minimum_DMA bit field. DMA latency increases for each microsecond an engine must wait.

The read and write DMA channels use FIFOs to buffer small amounts of PCI bus data. The FIFOs provide elasticity for data movement between internal memory and the PCI interface. Host software may configure DMA watermarks—values where PCI activity is enabled/disabled (see Figure 84).

When enqueued data is less than the watermark value, PCI bus transactions are inhibited. The DMA channel will wait until the FIFO fills above the threshold before initiating PCI transactions. Host software may configure the DMA_Write_Watermark bit fields to set the activity threshold in the write FIFO. The DMA_Write_Watermark bit field is read/write and is also located in the DMA Read/Write register. The DMA_Read_Watermark is located in the DMA Read/Write register and sets the activity threshold in the read FIFO. The DMA_Read_Watermark bit field is read/write. Both the read and write watermark registers default to zero after power-on reset.

When the BCM5704 performs a DMA read to access a buffer descriptor that ends exactly on any multiple of 4 GB, the BCM5704 will generate a Host Address Overflow Error, even though the buffer descriptor does not actually cross over the 4 GB boundary. To avoid this fatal error, the host driver must ensure that all buffer descriptors do not end on any multiple of 4 GB. This is not an issue for BCM5704 B0 and later.

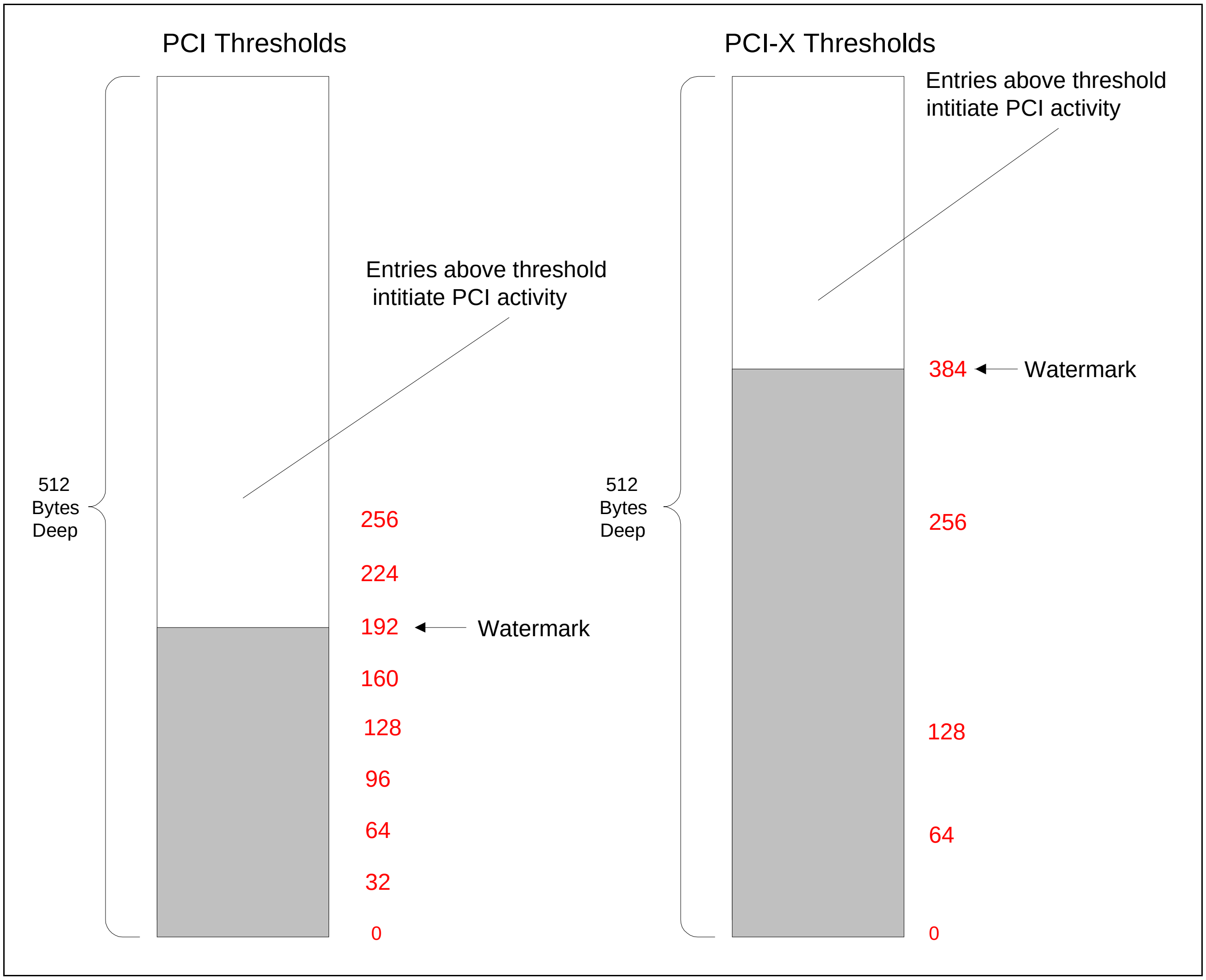


Figure 84: Watermark Levels for PCI and PCI-X Queued Data

Parity Errors

Bus masters are responsible for driving PAR and PAR64 on PCI transactions (see Figure 85). The PAR signal corresponds to AD[0..31] lines, and the PAR64 signal matches to the AD[32..63] lines. The BCM57XX family drives the parity signals on bus master initiated transactions. Driving the PAR and PAR64 signals is required per section 3.7.1 in the PCI v2.2 specification. A PCI target device will assert PERR when a data phase parity error is detected.

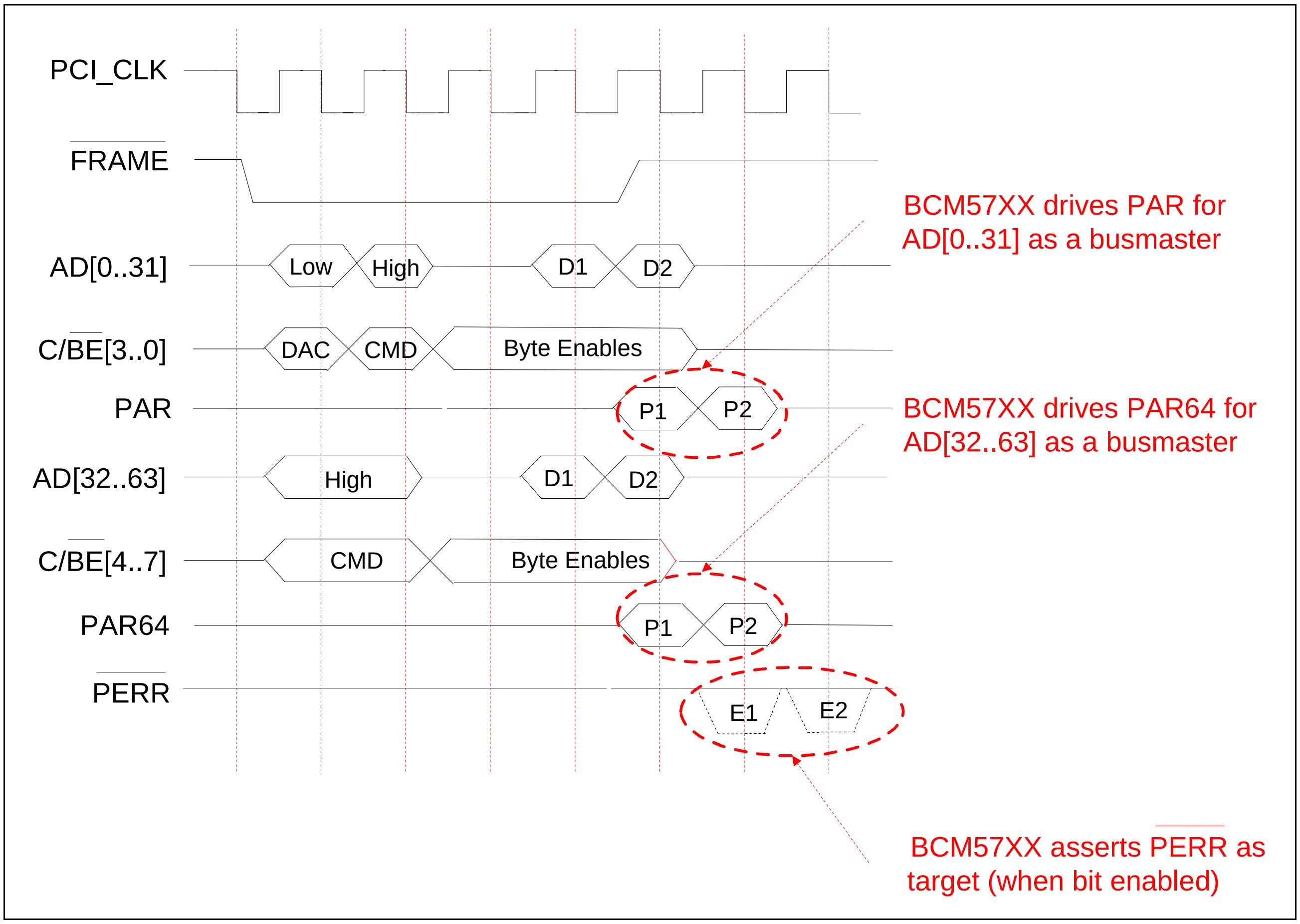

PCI_CLK
FRAME
AD[0..31]
Low
High
D1
D2
C/BE[3..0]
DAC
CMD
Byte Enables
PAR
P1
P2
AD[32..63]
High
D1
D2
C/BE[4..7]
CMD
Byte Enables
PAR64
P1
P2
PERR
E1
E2
BCM57XX drives PAR for AD[0..31] as a busmaster
BCM57XX drives PAR64 for AD[32..63] as a busmaster
BCM57XX asserts PERR as target (when bit enabled)

Figure 85: Bus Masters Drive PAR and PAR64 on PCI Transactions

Parity_Error_Enable is a read/write bit in the PCI configuration space Command register (see "Command Register (Offset 0x04)" on page 302). Host software should set this bit to enable the PCI/PCI-X interface reporting of parity errors for both BCM57XX PCI target claim and master initiated transactions. The BCM57XX family (target) will assert the PERR signal when the Parity_Error_Enable bit is enabled. This bit is disabled on reset. The target of a PCI transaction may assert the PERR signal. The BCM57XX family (target) will detect parity errors and assert the PERR signal if a data phase parity error is detected.

Software may still poll the PCI Configuration Space Status register (see "Status Register (Offset 0x06)" on page 303) and read the Detected_Parity_Error bit. This bit will always be set regardless of whether the BCM57XX family asserts the PERR signal. The Parity_Error_Enable bit enables/disables the PERR signal; whereas the status bit will always reflect parity error for target transactions. Software must explicitly clear this bit to differentiate a subsequent transaction.

The BCM57XX family (master) will assert the Master_Data_Parity_Error bit also in the PCI configuration space Status register when the transaction target asserts PERR. The BCM57XX family (master) is responsible for driving PAR and PAR64 during the data phase of a transaction. The target device can detect a parity error, then respond with a PERR assertion. The target will assert PERR when the PAR/PAR64 signals do not match the parity of the data driven on the AD[31:0]/AD[63:32] lines. Unlike the Detected_Parity_Errors bit, the Master_Data_Parity_Error bit will only be asserted when parity error detection is enabled—see Parity_Error_Detection bit in the PCI Command register (above).

Host software may query the Read_DMA_PCI_Parity_Error bit in the Read DMA Status register. This bit should be asserted when the BCM57XX family read channel encounters a parity error during a bus master read transaction. The assertion of this bit has a relationship to the assertion of the Master_Data_Parity_Error bit—both bits should assert on a read parity error. This bit reveals more details regarding the type of PCI transaction on which the parity error surfaced. Diagnostic software is an example host application interested in reading this bit.

The DMA write channel will also report parity errors to host software. The Write_DMA_PCI_Parity_Error bit is set when the DMA Write engine encounters a parity error during a write transaction to a target device. The Write_DMA_PCI_Parity_Error bit has a relationship to the Master_Data_Parity_Error bit; both bits are set on a write transaction parity error. The Write_DMA_PCI_Parity_Error bit is located in the Write DMA Status register and software must write to clear this bit. This bit defaults to zero after reset. This bit reflects parity errors for BCM57XX family (master) write transactions.

The Message Signaled Interrupt (MSI) portion of the MAC architecture uses the PCI interface to write message values to a host mailbox (see “MSI” on page 289). The MSI engine will record parity errors encountered during a PCI write or PCI-X write DWORD transaction. The PCI_Parity_Error bit in the MSI Status register (see “MSI Status Register (Offset 0x6004)” on page 499) reflects the detected parity error. The Master_Data_Parity_Error bit is also set when a bus master MSI write detects the parity condition. The PCI_Parity_Error bit is located in the MSI Status register, and software must write to clear this bit. The bit defaults to zero after reset.

A parity error and recovery feature is available when the MAC is operating in PCI-X mode. The BCM57XX family will assert the SERR signal when a parity error is detected. When the Data_Parity_Error_Recovery_Enable bit is set, the MAC will avoid the assertion of SERR. Host software may implement a recovery methodology. The BCM57XX architecture allows a host driver to reset the MAC, under this scenario. The BCM57XX family will not retry the transaction. The Data_Parity_Error_Recovery_Enable bit is located in the PCI-X Command register (see “PCI-X Command Register (Offset 0x42)” on page 312); this bit is read write and defaults to zero after reset.

Note: Most server chipsets are configured to assert a NMI when a parity error is first detected. The assertion of PERR on such chipsets allows the BCM57XX family to master terminate a transaction; however, a NMI will be generated regardless. A parity NMI may result in an O/S fault. For example, a blue screen under Microsoft® Windows. Parity recovery requires the coordination of many system software components: O/S, BIOS, and device drivers. The configuration of the Data_PCI_Parity_Error_Recovery_Enable bit must be coordinated as a system wide application. This configuration of this bit, by itself, is not sufficient to recover from PCI parity errors.

PCI-X Command Usage

A PCI-X transaction includes an attribute phase between the address and data phases of a read/write transaction (see Figure 86). The BCM57XX family will drive an attribute value to indicate that the transaction ordering may be relaxed. Reference section 2.5 of the PCI-X v1.0 specification for a discussion of the attribute bit assignments.

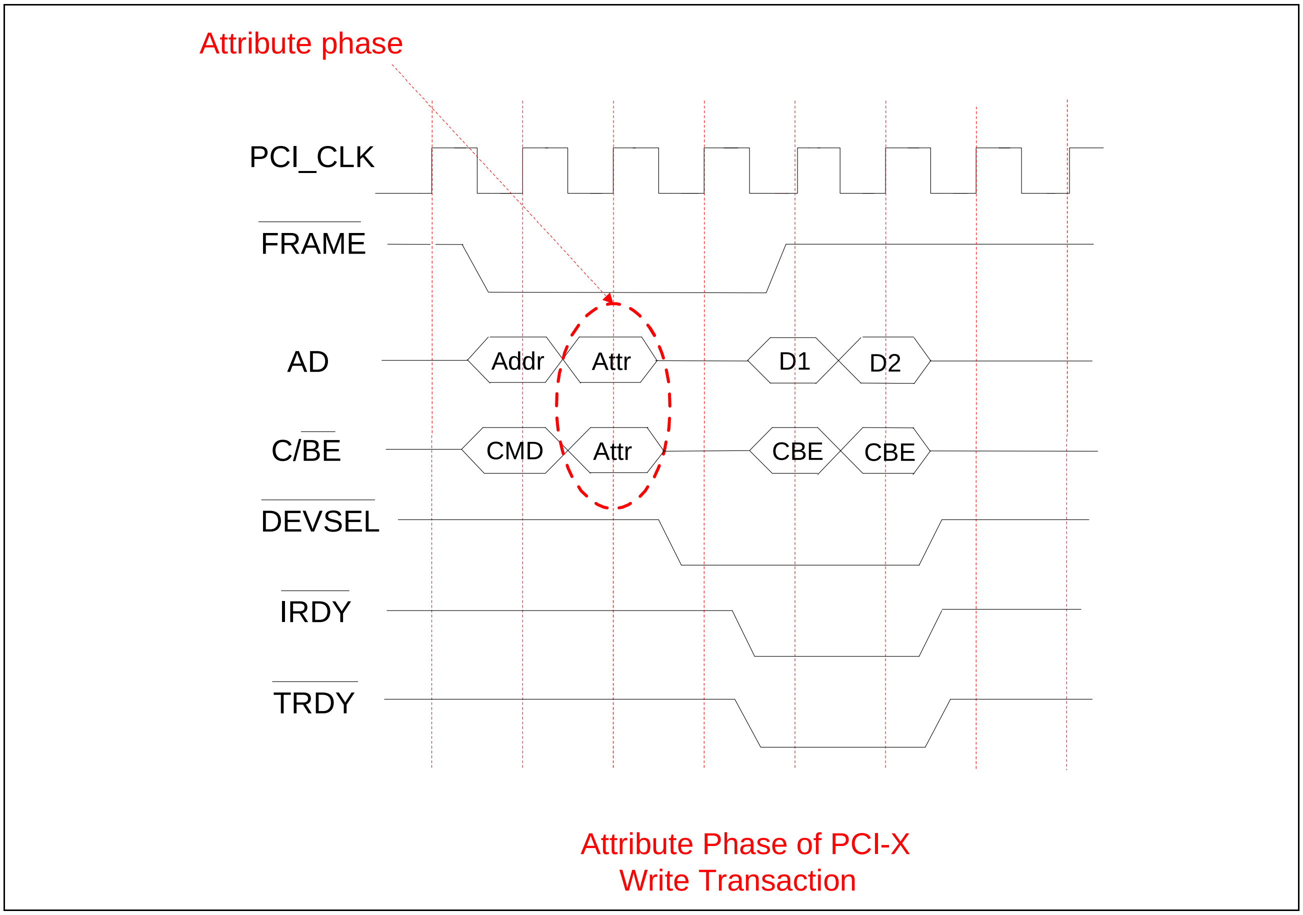


Figure 86: PCI-X Transaction Includes Attribute Phase

The PCI-X specification allows initiators to insert a relaxed order flag in the attribute portion of a transaction. The benefit of relaxed ordered transactions is not apparent on the PCI-X bus; rather the host local memory bridge gets a performance benefit from this setting. For write transactions, relaxed ordered transactions are allowed to pass ordered write transactions that have been posted to the host bridge. Host designs may have multiple memory controllers that service ranges of physical memory. Additionally, memory controllers can interleave local memory transactions amongst several DRAM banks. These mechanisms provide the host bridge multiple paths to physical memory. The relaxed order bit allow the host bridge to forward transactions without serialization in the posted write buffer. On a PCI-X bus, read transactions that have the relax order attribute may pass posted write transactions. A relax ordered read transaction must not reference an address that will be updated by a transaction in the posted write queue. The relax order read transaction must not have a relationship to any of the posted writes. There is significant performance degradation for memory reads that must wait for the completion of memory writes. Relax order reads loosen the restriction that all reads must serialize behind writes on the PCI bridge. Reference section 11.2 in the PCI-X v1.1 specification. See Figure 87.

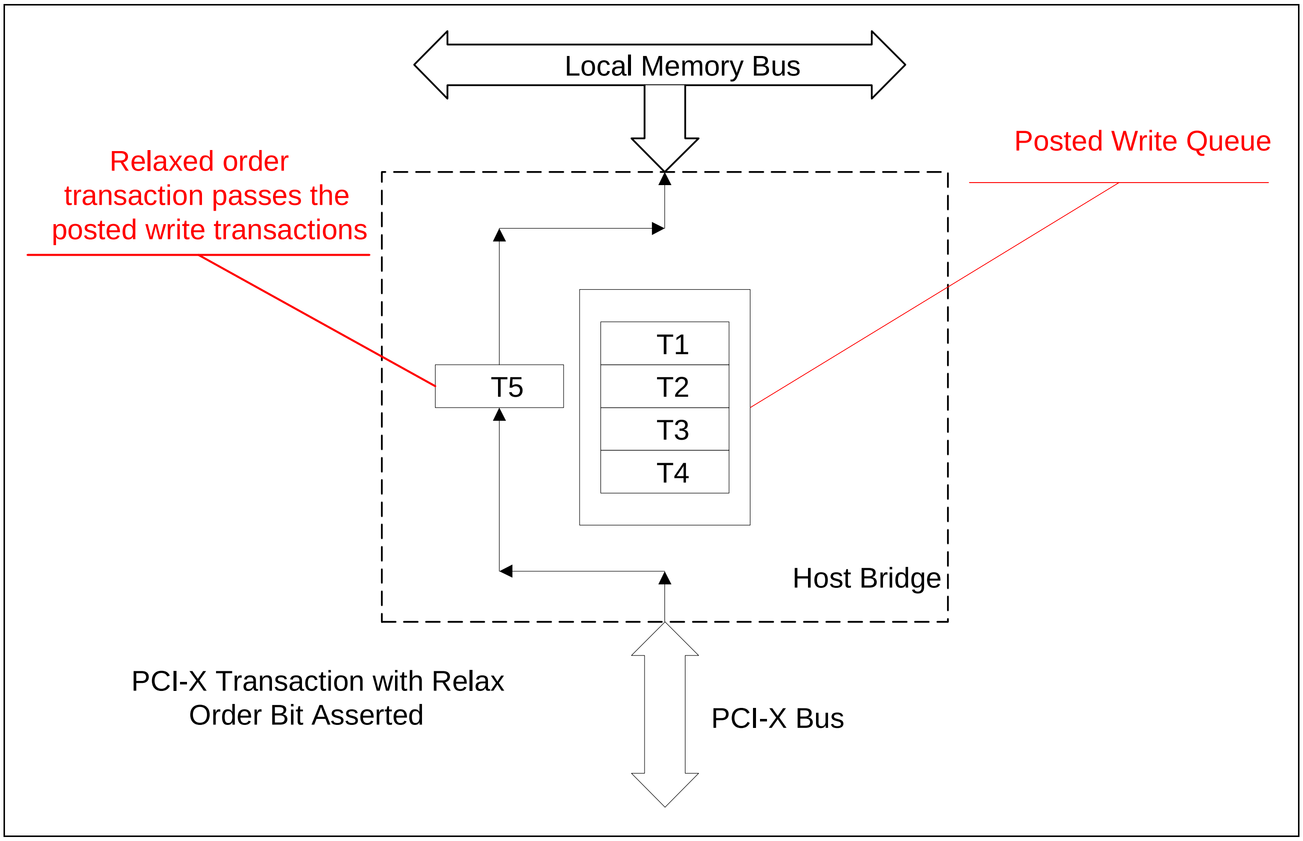
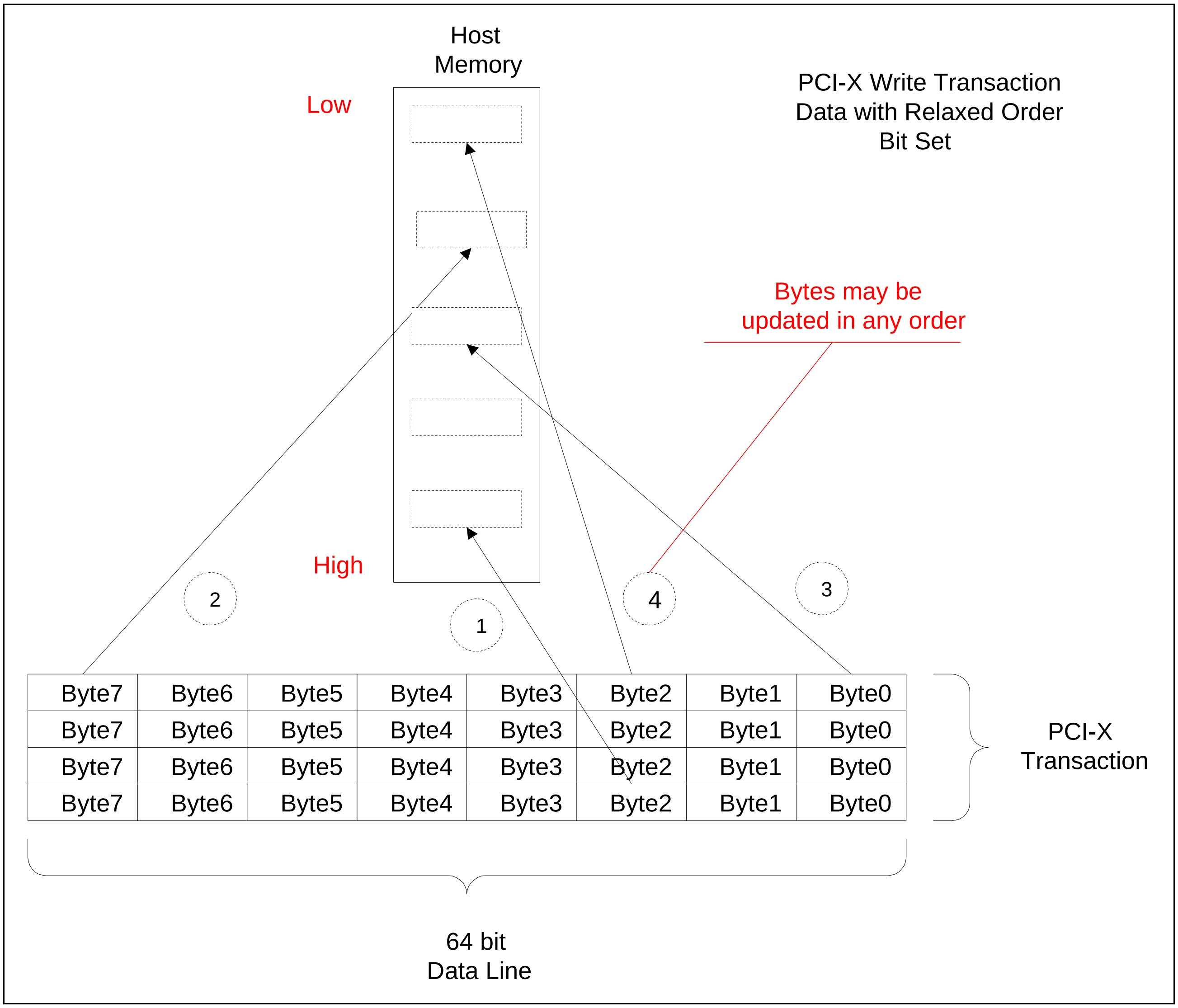

Local Memory Bus
Relaxed order transaction passes the posted write transactions
Posted Write Queue
T1
T2
T3
T4
T5
Host Bridge
PCI-X Transaction with Relax Order Bit Asserted
PCI-X Bus

Figure 87: Relaxed Ordered Transactions

Another benefit for PCI-X write transactions with the relaxed order attribute is out of order memory writes (see Figure 88). The memory controller may write PCI-X transaction data to physical memory in any order/sequence. The memory controller benefits from the ability to gather physical address ranges based on memory bank addressibility. The exact mechanism for the write ordering is left to the memory controller. The only requisite is that the bytes written must route to the correct address ranges/groups.



Byte7	Byte6	Byte5	Byte4	Byte3	Byte2	Byte1	Byte0
Byte7	Byte6	Byte5	Byte4	Byte3	Byte2	Byte1	Byte0
Byte7	Byte6	Byte5	Byte4	Byte3	Byte2	Byte1	Byte0
Byte7	Byte6	Byte5	Byte4	Byte3	Byte2	Byte1	Byte0

Figure 88: Out-of-Order Memory Writes

The Enable_Relaxed_Ordering bit in the PCI-X command register (see "PCI-X Command Register (Offset 0x42)" on page 312) will set/clear the relaxed order bit in the attribute portion of PCI-X transactions. The bit defaults to enabled, after the device reset. Host software may read and write the Enabled_Relaxed_Ordering bit.

Note: The BCM57XX family does not differentiate PCI-X write transactions from the host status block and packet buffer memory. If the Enable_Relaxed_Ordering bit is enabled, the PCI-X interface may tag both the status and packet write transactions with the relaxed ordering attribute. Consequently, the status block transactions may not complete before packet buffer DMA. System software may read stale under this scenario. System software should clear the Enable_Relaxed_Ordering bit during initialization. This bit should not be enabled on the BCM57XX family.

Bit Name	Required Value
Enable_Relaxed_Ordering	0 (Disabled)

The PCI-X interface may insert the No Snoop attribute into PCI-X transactions. Host processors will snoop transactions on the front side bus and then update their local caches. A master-initiated transaction from the NIC to host memory controller will be snooped by the host processor. When the No Snoop attribute is present, the range of addresses within the transaction should not be stored in a processor's cache. The host software/driver must typically ensure that the address range is not cached through a buffer allocation or flush call. The No_Snoop bit in the PCI_State register (see "PCI State Register (Offset 0x70)" on page 332) is read/write and this bit defaults to zero after reset. If this bit is asserted, the NS attribute in the PCI-X transaction is asserted. Refer to Section 2.5 in the PCI-X v1.0 specification.

Note: The effectiveness of the No_Snoop bit is dependent upon the processor front side bus architecture. Architectures, such as MESI, maintain cache coherency between system memory and the L2 caches. Any update from a PCI-X master to main memory will be snooped by the processors and the cache line updated accordingly. The No Snoop bit has no influence in architectures where host memory is always cache coherent. Read/write transactions from the PCI-X master will always be snooped to prevent an incoherency error on the local processors.

The BCM57XX family can initiate memory read bursts on the PCI-X bus. The PCI-X interface may use commands such as Memory Read Block when initiating a transaction with a target memory device. These commands allow the target device to stream data back to the master and increase bus utilization. Overhead is associated with every PCI or PCI-X transaction; bus arbitration, address decode, and attribute phases all use PCI-X clock cycles. Each clock cycle that spent with protocol overhead reduces the effective bandwidth of the bus. A read data burst minimizes the protocol overhead associated with every memory read. The obvious downside to long data bursts is device latency—other peripherals must wait for bus access. The Maximum_Memory_Read_Byte_Count bit in the PCI-X Command register is read/write and default to a value of zero after reset.

Note: The BCM57XX family is only capable of 512 byte data bursts; host software must set the maximum memory read count for the BCM57XX family accordingly. Settings greater than 512 bytes are not valid for the BCM57XX family.

Bit Name	Required Value
Maximum Memory Read Byte Count	0

Note: For BCM5703C/BCM5703S in PCI-X mode, the DMA Read Watermark bits in DMA Read/Write Control register (see "DMA Read/Write Control Register (Offset 0x6C)" on page 327) should be programmed to less than or equal to Maximum Read Byte Count.

Max Read Byte	*DMA Read Watermark (Allowed Values)*
00 (512 bytes)	000: 64 Bytes 001: 128 Bytes 010: 256 Bytes 011: 384 Bytes 100: 512 Bytes
01 (1024 Bytes)	000: 64 Bytes 001: 128 Bytes 010: 256 Bytes 011: 384 Bytes 100: 512 Bytes 101: 1024 Bytes

The BCM57XX family supports PCI-X split read transactions. See "PCI-X Host Bus Interface (Applicable to BCM5700, BCM5701, BCM5703C, BCM5703S, BCM5704C, and BCM5704S)" on page 83. Host software may program the Maximum_Outstanding_Split_Transaction bit field in the PCI-X Command register with a value that reflects the number of outstanding split memory reads. This register is read/write and defaults to zero after reset. System software may tune PCI-X performance using the Maximum_Outstanding_Split_Transaction field. The PCI-X interface does not support split memory write transactions; therefore, this register will only reflect the total number of split memory reads outstanding.

Note: The BCM57XX family can handle one outstanding split memory read. The DMA write engine may have a PCI-X write transaction outstanding, but it will not be a split transaction. Two total transactions may be outstanding, but only the DMA read engine may have a split transaction outstanding. Host software should not modify this bit field for the BCM57XX family.

Bit Name	*Required Value*
Maximum Outstanding Split Transactions	0

If an error occurs on the PCI-X bus during a split memory read transaction, the Received_Split_Completion_Error_Message bit in the PCI-X Status register (see "PCI-X Status Register (Offset 0x44)" on page 314) will be asserted. This bit is write to clear and will be initialized with zero after device reset. An error condition is returned in the attribute portion of the PCI-X transaction. Refer to section 5.4.6 of the PCI-X v1.0 specification for error codes and class information. System (diagnostic) software may read the Received_Split_Completion_Error_Message bit to determine if a split completion error has been detected by the MAC.

If system (diagnostic) software detects a split completion error by reading this flag, further information can be determined by reading class and index error codes in the PCI state register. Table 5-1 in the PCI-X v1.1 specification contains the error message class and index codes; refer to the specification for the latest data. The Split_Completion_Message_Class bit field contains a value that identifies the class of split completion error. The errors are grouped/categorized by a class type. Within each class, there are instances/types of errors. The Split_Completion_Message_Index bit field contains a value for a specific type of split completion error—an instance in the error class. Both bit fields are located in the PCI State register (see "PCI State Register (Offset 0x70)" on page 332).

Register Quick Cross Reference

BCM57XX Family

The following table lists the BCM57XX PCI-X registers.

Table 92: PCI -X Registers

Register	*Bit*	*Cross Reference*
DMA Read/Write	Use_MemRdMult_Command	See "DMA Read/Write Control Register (Offset 0x6C)" on page 327.
DMA Read/Write	Default_PCI_Write_Command	
DMA Read/Write	Default_PCI_Read_Command	
DMA Read/Write	Assert_All_BE_On_DMA_Write	
DMA Read/Write	DMA_Read_Watermark	
DMA Read/Write	DMA_Write_Watermark	
DMA Read/Write	DMA_Read_Address_Boundary	
DMA Read/Write	DMA_Write_Address_Boundary	
DMA Read/Write	One_DMA_At_Once	
DMA Read/Write	Minimum_DMA	
PCI Command	Bus_Master	See "Command Register (Offset 0x04)" on page 302.
PCI Command	Memory_Write_And_Invalidate	
PCI Command	Parity_Error_Enable	
PCI Status	Detected_Parity_Error	See "Status Register (Offset 0x06)" on page 303.
PCI Status	Master_Data_Parity_Error	
Write DMA Status	Write_DMA_PCI_Parity_Error	See "Write DMA Status Register (Offset 0x4C04)" on page 482.
Read DMA Status	Read_DMA_PCI_Parity_Error	See "Read DMA Status Register (Offset 0x4804)" on page 479.
MSI Status	PCI_Parity_Error	See "MSI Mode Register (Offset 0x6000)" on page 498.
PCI-X Command	Data_Parity_Error_Recovery_Enable	See "PCI-X Command Register (Offset 0x42)" on page 312.
PCI-X Command	Maximum_Outstanding_Split_ Transaction	
PCI-X Command	Maximum_Memory_Read_Byte_Count	
PCI-X Command	Enable_Relaxed_Ordering	
PCI-X Status	Received_Split_Completion_Error_ Message	See "PCI-X Status Register (Offset 0x44)" on page 314.
PCI State	Conventional_PCI	See "PCI State Register (Offset 0x70)" on page 332.
PCI State	33/66PCI_66/133PCIX_Mode	
PCI State	No_Snoop	
PCI State	Split_Completion_Message_Class	
PCI State	Split_Completion_Message_Index	

EXPANSION ROM

DESCRIPTION

The expansion ROM on the BCM57XX family is intended for implementation of PXE (Preboot Execution Environment). The BCM5700 and BCM5701 support up to 64 KB of Expansion ROM, while the other members of the BCM57XX family support up to 16 MB of Expansion ROM.

OPERATIONAL CHARACTERISTICS

By default, the Expansion ROM is disabled and the firmware has to explicitly enable this feature by setting PCI_State.PCI_Expansion_ROM_Desired bit to one (see "PCI State Register (Offset 0x70)" on page 332). Once this bit is enabled, the BCM5700 and BCM5701 devices of BCM57XX NetXtreme family will map Expansion ROM space to Internal Memory from 0x10000-0x1ffff (see Figure 89). In other words, if the BIOS reads the first byte of the Expansion ROM, then the BCM57XX family will return the first byte at location 0x10000 of Internal Memory. In rest of the BCM57XX NetXtreme devices, the bootcode firmware handles the Expansion ROM accesses of the device.

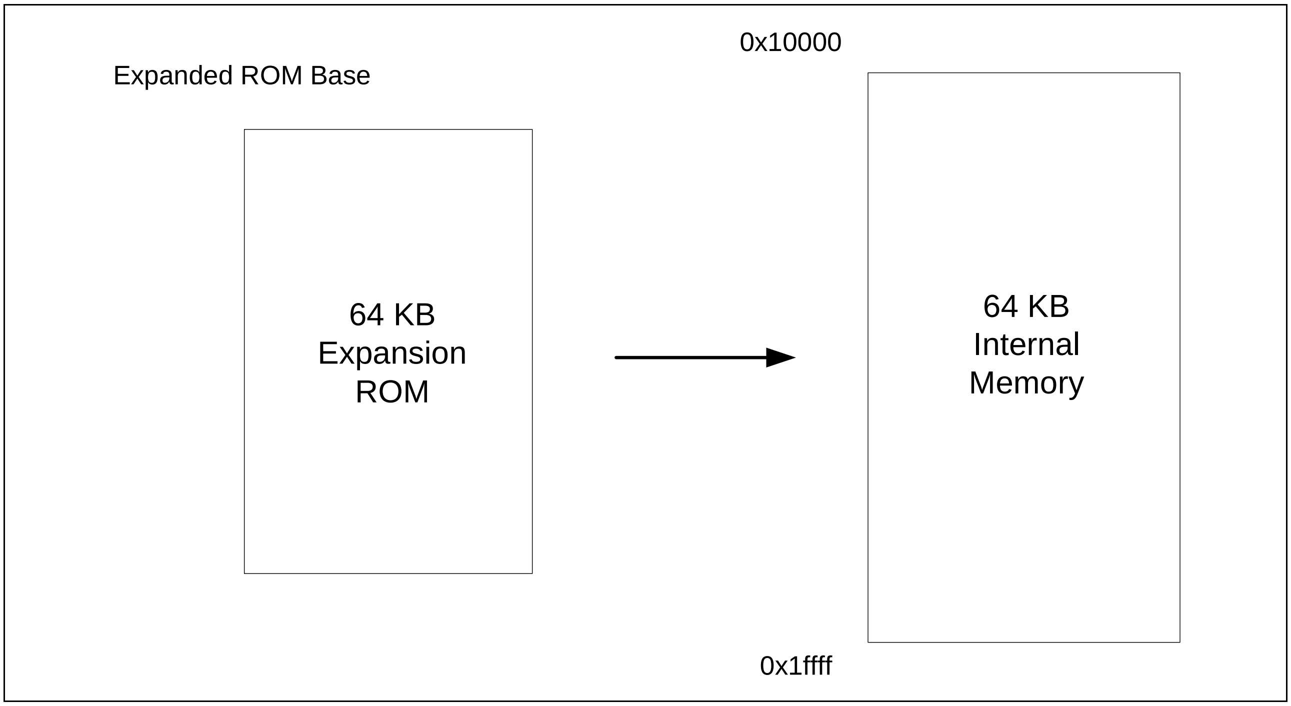


Figure 89: Mapping Expansion ROM Space into Internal Memory (BCM5700 & BCM5701 only)

BIOS

The BIOS detects if a PCI device supports Expansion ROM or not by writing the value 0xFFFFFFFE to Expansion_ROM_Base_Register (Register 0x30 of PCI Configuration). The BIOS then reads back from this register. If the value is nonzero, then this PCI device supports Expansion ROM; otherwise, it does not. In the case of the BCM57XX family, the BCM5700 and BCM5701 returns a value of 0xFFFF0000 if the PCI_State.PCI_Expansion_ROM_Desired bit is set. This indicates either a 64-KB expansion ROM, or a value appropriate for the expansion ROM size selected in NVRAM (see "NVRAM Configuration" on page 88) for the other members of the BCM57XX family. On the other hand, if the PCI_Expansion_ROM_Desired bit cleared, then the BCM57XX family will return a value of 0x00000000. This indicates to the BIOS that no Expansion ROM is supported.

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 two bytes, and so forth) 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.

PXE (Preboot Execution Environment)

PXE is implemented as an Expansion ROM in the NIC implementation. In the LOM implementation, PXE should reside in the system BIOS. In the NIC implementation, PXE image is stored in the NVRAM. Upon power on reset of the BCM57XX family, the RX RISC will load the boot code from the NVRAM into RX RISC scratchpad and execute. This boot code will program the device with programmable manufacturing information (such as MAC address, PCI vendor ID/device ID, etc.). If PXE is enabled, the boot code will read PXE image from the NVRAM and write to the Internal Memory of the BCM5700/BCM5701 starting at 0x10000.

Reset and Timing Considerations (BCM5700 and BCM5701 Only)

Since the Expansion ROM image (up to 64K bytes) is stored in the NVRAM, the amount of time to load this image to Internal Memory can be up to 3 or 4 seconds. This is due to slowness of the NVRAM device. Based on the PCI specification, the PCI device should not be accessed in 2^{25} clock cycles. If the PCI clock is 133 MHZ, the device can be accessed in 250 ms. It can be a potential problem if the BIOS accesses Expansion ROM when the Expansion ROM image is not completely copied to Internal Memory by the boot code firmware. The BCM57XX family provides a way to avoid this problem by forcing PCI retry if the Expansion ROM is accessed when the content of Expansion ROM is not ready. The details of solving this problem are described as follows:

1. The the RX RISC will load the boot code from the NVRAM into RX RISC scratchpad and execute after the BCM57XX family is reset. It takes about 20 to 50 ms (depends on the boot code size) before the first instruction of the boot code is executed.
2. When PXE is enabled, the boot code first sets PCI_State, PCI_Expansion_ROM_Retry, and PCI_State.PCI_Expansion_ROM_Desired to 1 before moving the Expansion ROM image from the NVRAM to internal memory.
3. If the BIOS accesses Expansion ROM during this process, the BCM57XX family will force PCI retry on this access.
4. When the Expansion ROM image is completely moved to Internal Memory, the boot code will clear PCI_State.PCI_Expansion_ROM_Retry to zero.

Boot code is executed whenever the BCM57XX family is reset via PCI Reset or S/W device reset. PXE initialization should only be necessary after a PCI reset. The boot code differentiates PCI Reset or S/W device reset by checking content in Internal Memory at 0xb50 when it is loaded and executed. If the content is 0x4B455654, then it is due to S/W Reset. Therefore, the device driver has to initialize 0xb50 with 0x4B455654 before issuing a S/W device reset.

Power Management

Description

The BCM57XX family is compliant with the PCI v1.1 power management specification. The MAC is programmable to two ACPI states: D0 and D3. The D0 state is a full power, operational mode—all the MAC core functions run at the highest clocking frequency, and components are fully functional. The MAC may be either initialized or un-initialized in the D0 ACPI state. An un-initialized D0 state is entered through a device reset or PME event; the MAC functional blocks are not started and initialized. Host software must reset/initialize hardware blocks to transition the device to a D0 initialized (active) state. The D0 active state places the device into a full power/operational mode. Receive and transmit data paths are fully operational, and the PCI block is initialized for bus mastering DMA.

Host device drivers do not differentiate between D3 hot and D3 cold states. ACPI compliant device drivers are unloaded and quiescent in the D3 state and PCI slot power state is transparent. When the MAC is in D3 hot state, PCI slot power (3.3V or 5.0V) is available to power the PCI I/O pins. The PCI configuration and memory space may be accessed in D3 hot state. The core clock must remain enabled, so the MAC can respond to PCI configuration and memory transactions. The Disable_Core_Clock bit, in the PCI Clock Control register (see "PCI Clock Control Register (Offset 0x74)" on page 334) enables/disables clocking in the core clock domain. A D3 cold state provides only the PCI Vaux supply—PCI slot power is not present. The MAC will consume a maximum of 375 mA in a D3 cold power management mode.

The following functional blocks are integral to MAC power management:

- PMSCR register
- PCI Clock Control register
- Miscellaneous Control register
- WOL
- PCI Vaux Supply
- PCI Slot Power Supply
- GPIO

Operational Characteristics

Figure 90 applies to the BCM57XX family reference designs. The MAC GPIO pins are available for application specific usage; however, Broadcom encourages both software and hardware engineers to follow the Broadcom design guidelines and application notes. NIC and LOM designs based on the BCM57XX family controller family use external board level logic to switch power regulators for D3 ACPI mode. GPIO pins 0, 1, and 2 are all used for external power switching and are programmed by host software.

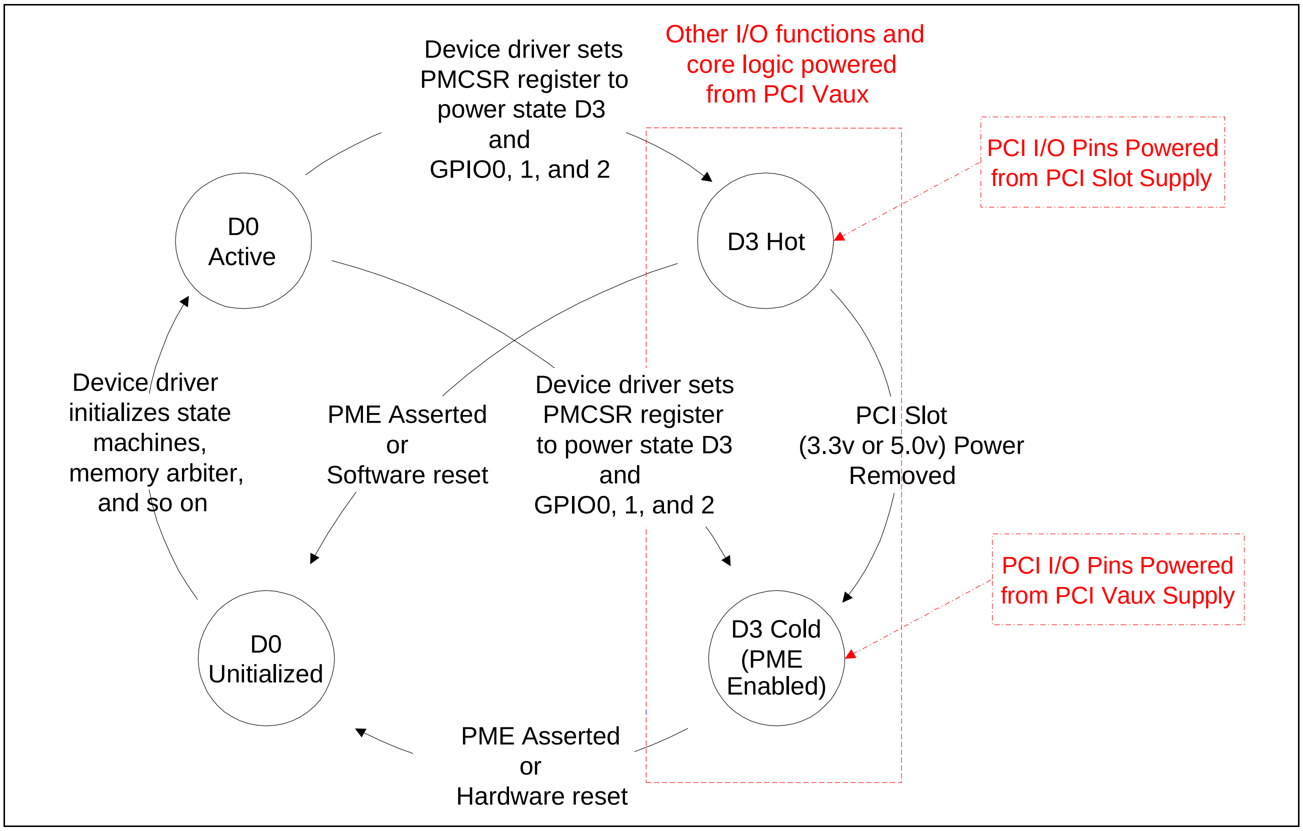


Figure 90: Power State Transition Diagram

Device State D0 (Uninitialized)

The D0 state is entered after a PCI reset or device (software) reset. The assertion of $\overline{\text{PME}}$ will cause the PCI bridge to drive $\overline{\text{RST}}$. The MAC hardware blocks are not initialized in this state. For example, the RX engine, TX engine, multicast filter, and memory arbiter are all uninitialized. All the MAC functional blocks are powered.

Device State D0 (Active)

Host software has initialized the MAC hardware blocks. The RX and TX data paths are ready to send/receive Ethernet packets. The PCI block is available to DMA packets to host memory. This is a full power ACPI operational state. Host software/drivers must follow the initialization procedure (see "Initialization" on page 146) to move the MAC into a D0 active state.

When the BCM5705, BCM5721, and BCM5751 MAC Transceivers detect that main power is lost and it's still in the D0 state, it will reset itself to the D3 (Cold) state and then operate in 10/100 mode, like the OOB WOL state.

Device State D1

Supported on BCM5705, BCM5721, and BCM5751 MAC Transceivers only.

Device State D2

Supported on BCM5705, BCM5721, and BCM5751 MAC Transceivers only.

Device State D3 (Hot)

The MAC's configuration space and memory mapped I/O blocks are accessible in D3 hot state. The PCI I/O drivers are still powered by slot power in this state. However, host software has switched the MAC to use PCI Vaux for VDD_CORE and VDD_IO. GPIO pins 0, 1, and 2 are configured before the transition to this state. The RX and TX RISC processor clocks have been stopped in this state. The core clock remains active so PCI transactions may be processed by the MAC (see "PCI Clock Control Register (Offset 0x74)" on page 334). This is a low-power state where some key components have been powered down. The physical layer auto-advertises 10 Mbps capability in this state, and link is set to 10 Mbps half or full-duplex. The PHY is configured for WOL mode. WOL pattern filters are initialized and active; the MAC will process Magic Packets™. The host chipset implements the power management policy for the PCI bus; the MAC driver does not influence the PCI Vaux nor Slot power supply.

Note: For BCM5751 and BCM5721 devices with version A1 or later, the drivers should use configuration cycles (not the memory write cycles) to write to the PMCSR register at offset 0x4C for putting the device in D3 Hot state.

Device State D3 (Cold)

The MAC is completed powered by PCI Vaux in D3 cold. PCI configuration space and memory mapped I/O are not available. The only portion of the MAC active is the WOL pattern and Magic Packet filters[2]. The MAC will assert a $\overline{\text{PME}}$ in this state and indicate to the host bridge that a wake up event has occurred. The host bridge will normally provide PCI Slot power and then reset the device. GPIO pins 0, 1, and 2 are configured the same as D3 hot. Host software does not differentiate between D3 hot/cold. The MAC and PHY will not consume more than 375 mA in this mode. The BCM5401 PHY must negotiate for 10 Mbps half/duplex speed. The PHY WOL mode is configured.

Note: The BCM57XX PCIe devices support the PCIe power management which is compatible with PCI bus power management.

[2] Magic Packet™ is a registered trade mark of AMD.

WAKE ON LAN

See "Wake-on LAN Mode/Low-Power" on page 268.

GPIO

The use of GPIO pins for power management is design-specific, though Broadcom-delivered drivers use GPIO pins in the manner listed in Table 93 and Table 94. This usage is only applicable when the BCM57XX is configured for a NIC design as specified in "Revision Levels" on page 5; it is not applicable to LAN-on-Motherboard (LOM) designs.

Table 93: GPIO Usage for BCM5700/BCM5701 Power Management for Broadcom Drivers

Function	Description	GPIO0	GPIO1	GPIO2
VAux	Switches BCM5700/BCM5701 to use VAux power	1	1	0

Table 94: GPIO Usage for BCM5703C/BCM5703S and Later Power Management for Broadcom Drivers[a]

Function	Description	GPIO0	GPIO1	GPIO2
VAux	Sequence for switching to VAux	0	0	1
		1	1	1
		1	1	0
VMain	Sequence for switching to VMain	x	1	x
		x	0	x
		x	1	x

[a] x = Don't Care

POWER SUPPLY IN D3 STATE

Table 95 shows the power supply to various power pins on the BCM57XX family. This table assumes that host software has switched power regulators using GPIO pins 0, 1, and 2.

Table 95: BCM57XX **Power Pins**

Pin	D3 Normal	D3 Hot	D3 Cold
VDD_CORE	PCI Slot	Vaux	Vaux
VDD_IO	PCI Slot	Vaux	Vaux
VDD_IO-PCI	PCI Slot	PCI Slot	No Power

CLOCK CONTROL

Certain functional blocks in the MAC architecture should be powered down before a transition to D3 ACPI state. MAC clock generators/PLLs drive transistor level logic, which switch states on every clock pulse. Transistor level switching consumes power (milliwatts). Software should selectively disable clocking to non-essential functional blocks. Software must set the Enable_Clock_Control_Register bit in the Miscellaneous Host Control register; the assertion of this bit allows host software to configure the PCI clock control register. The following clock bits should be configured in the PCI Clock Control register (see "PCI Clock Control Register (Offset 0x74)" on page 334):

- TX RISC clock disable
- RX RISC clock disable
- Select alternate clock—the 133-MHz PLL is not used as reference clock.

DEVICE ACPI TRANSITIONS

Host software must program the Power Management Control/Status (PMSCR) register to transition the device between D0 and D3 ACPI states. The Power State bit field in the PMSCR (see "Power Management Control/Status Register (Offset 0x4C)" on page 318) may be programmed to D0, D1, D2, and D3 states.

Note: The D1 and D2 configurations are not supported in the BCM57XX family. The D1 and D2 bit configurations are available for applications, where D1 and D2 states are introduced for board level designs—the bits provide flexibility to the application. The Broadcom reference NIC/LOM designs do not use D1 and D2 states; therefore, host software should avoid setting these states. Before the Mac is moved into the D3 state, the clocks and GPIO must be configured (see above sections).

The PME signal is enabled in the PMSCR by asserting the PME_Enable bit. Device drivers/BIOS may also read the PME_Status bit to determine whether the event has been driven; PME_Status is a write to clear bit. The type and supported power management features for the BCM57XX family are reported in the Power Management Capabilities (PMC) register (see "Power Management Capabilities Register (Offset 0x4A)" on page 317). System software and BIOS may read this register to enumerate and detect the power management features supported by the NIC/LOM. For example, the BCM57XX family can assert PME from both D3 hot and cold states. The PME_Support bit field in the PMC register will reflect this capability.

DISABLE DEVICE THROUGH BIOS

The BCM5721 and BCM5751 devices with boot code v3.22 or later can be disabled through BIOS by writing the value of DEADDEADh to shared memory location of B50h. This eliminates the need for BIOS to execute the device specific procedure for disabling the MAC controller device. The BIOS must do the following steps to disable the BCM5751 device.

1. Config cycle, write 88h to location 68h.
2. Config cycle, write 0B50h to location 7Ch.
3. Config cycle, write DEADDEADh to location 84h.

REGISTER QUICK CROSS REFERENCE

The BCM57XX power management registers are listed in Table 96.

Table 96: BCM57XX ***Power Management Registers***

Register	*Bit*	*Description*	*Cross Reference*
Misc Local Control	Misc_Pin_0_Output	GPIO pin 0	See "Miscellaneous Local Control Register (Offset 0x6808)" on page 507.
Misc Local Control	Misc_Pin_0_Output_Enable	When asserted, MAC drives pin output	
Misc Local Control	Misc_Pin_1_Output	GPIO pin 1	
Misc Local Control	Misc_Pin_1_Output_Enable	When asserted, MAC drives pin output	
Misc Local Control	Misc_Pin_2_Output	GPIO pin 2	
Misc Local Control	Misc_Pin_2_Output_Enable	When asserted, MAC drives pin output	
Misc Host Control	Enable_Clock_Control_Register		
PCI Clock_Control	TX RISC_Clock_Disable	Disable the clock to the transmit CPU	See "PCI Clock Control Register (Offset 0x74)" on page 334.
PCI Clock_Control	RX RISC_Clock_Disable	Disable the clock to the receive CPU	
PCI Clock_Control	Select_Alternate_Clock	Use an alternate clock as a reference, rather than the PLL 133	
PCI Clock_Control	PLL133	Disable the 133 MHz phased locked loop	
PCI Power Management Control/Status	PME_Enable		See "Power Management Control/Status Register (Offset 0x4C)" on page 318.
PCI Power Management Control/Status	Power_State		

BYTE SWAPPING

BACKGROUND

There are two basic formats for storing data in memory — little-endian and big-endian. The endianess of a system is determined by how multibyte quantities are stored in memory. A big-endian architecture stores the most significant byte at the lowest address offset while little-endian architecture stores the least significant byte at the lowest address offset.

For example, the 32-bit hex value 0x12345678 would be stored in memory as shown in the following table.

Table 97: Endian Example

Address	***00***	***01***	***02***	***03***
Big Endian	12	34	56	78
Little Endian	78	56	34	12

Another way of viewing how this data would be stored is shown in the following tables.

Table 98: Storage of Big-Endian Data

Storage Byte	***00***	***01***	***02***	***03***
Data Contents	12	34	56	78

Table 99: Storage of Little-Endian Data

Storage Byte	***03***	***02***	***01***	***00***
Data Contents	12	34	56	78

Examples of big-endian platforms include SGI Irix, IBM RS6000, and SUN.

Examples of little-endian platforms include Intel x86 and DEC Alpha.

PCI assumes a little-endian memory model. PCI configuration registers are organized so that the least significant portion of the data is assigned to the lower address.

ARCHITECTURE

The BCM57XX family is internally a big-endian machine, and its internal processors are big-endian devices. The BCM57XX family stores data internally in big-endian format using a 64-bit memory subsystem.

However, many hosts (e.g., x86 systems) use the little-endian format, and the PCI bus uses the little-endian format. Therefore the BCM57XX family has a number of byte swapping options that may be configured by software so that Little or Big Endian hosts can interface as seamlessly as possible with BCM57XX family over PCI. The BCM57XX family has the following bits that control byte and word swapping:

- **Enable Endian Word Swap** (bit 3, Miscellaneous Host Control register (offset 0x68 into PCI Config register). If 1, this register enables 32-bit word swapping when accessing the BCM57XX family via the PCI target interface.
- **Enable Endian Byte Swap** (bit 2, Miscellaneous Host Control register (offset 0x68 into PCI Config register). If 1, this register enables byte swapping (within a 32-bit word) when accessing the BCM57XX family via the PCI target interface.
- **Word Swap Data** (bit 5, Mode Control register (offset 0x6800 into the BCM57XX registers). If 1, this register enables word swapping of frame data when it comes across the bus.
- **Byte Swap Data** (bit 4, Mode Control register (offset 0x6800 into the BCM57XX registers). If 1, this register enables byte swapping of frame data when it comes across the bus.
- **Word Swap Non-Frame Data** (bit 2, Mode Control register (offset 0x6800 into the BCM57XX registers). If 1, this register enables word swapping of non frame data (i.e,. buffer descriptors, statistics, etc.) when it comes across the bus.
- **Byte Swap Non-Frame Data** (bit 1, Mode Control register (offset 0x6800 into the BCM57XX registers). If 1, this register enables byte swapping of non frame data (i.e., buffer descriptors, statistics, etc.) when it comes across the *bus*.

The setting of the above swapping bits will affect the order of how data is represented when it is transferred across PCI. Since byte swapping is a confusing subject, examples will be shown that reflect how each byte swapping bit works

ENABLE ENDIAN WORD SWAP AND ENABLE ENDIAN BYTE SWAP BITS

The Enable Endian Word Swap, and Enable Endian Byte Swap bits affect whether words or bytes are swapped during target PCI accesses. Thus, these bits affect the byte order when the host is directly reading/writing to registers or control structures that are physically located on the BCM57XX family. These bits do not affect the byte ordering of packet data or other structures that are mastered (DMAed) by the BCM57XX family.

When the BCM57XX family is accessed via PCI (which is little endian) as a PCI target, the BCM57XX family must implicitly map those accesses to internal structures that use a 64-bit Big Endian architecture. In the default case where no swap bits are set the BCM57XX family maps PCI data to internal structures shown in Figure 91 and Figure 92.

	MSB							LSB
Internal Byte #	0	1	2	3	4	5	6	7
Internal Bit #	63	48	47	32	31	16	15	0
Example Content	88	89	8A	8B	8C	8D	8E	8F

	MSB							LSB
PCI Byte #	7	6	5	4	3	2	1	0
PCI Bit #	63	48	47	32	31	16	15	0
Example Content	88	89	8A	8B	8C	8D	8E	8F

Figure 91: Default Translation (No Swapping) on 64-bit PCI

Internal Byte Ordering

	31	16	15	0
0x00	88	89	8A	8B
0x04	8C	8D	8E	8F

PCI Byte Ordering

31	16	15	0	
8C	8D	8E	8F	0x00
88	89	8A	8B	0x04

Figure 92: Default Translation (No Swapping) on 32-bit PCI

As illustrated above, because the BCM57XX family uses an internal 64-bit big endian architecture, it will map (by default) the most significant byte of an 8-byte (64-bit) internal quantity to the most significant byte on a 64-bit PCI bus. This works nicely for quantities (fields) that are 64 bits in size (e.g., a host physical address). However, this can be confusing for quantities that are 32 bits in size. Without Word Swapping enabled, the host could easily access the wrong 32-bit quantity when making a 32-bit access.

Take, for example, a Ring Control Block (RCB). RCBs are on-chip structures and read/written by the host via PCI target accesses. The table below shows the big-endian layout of an on-chip RCB:

Table 100: RCB (Big Endian 32-bit format)

Byte #	0	1	2	3	
Bit #	31	16	15	0	
	MSB	Host Ring Address			0x00
				LSB	0x04
	MSB	MAX_Len LSB	Flags		0x08
	NIC Ring Address				0x0C

If Word Swapping is not enabled, and the host made a 32-bit read request to address 0x08, the four bytes of data returned on the PCI bus would actually be the NIC Ring Address rather than the Max_Len and Flags fields. This initially might seem counter-intuitive, but is explained in Figure 92 on page 233. Therefore, if a software driver running on an x86 host (Little Endian) referenced on-chip data structures as they are defined in the BCM57XX data sheet, the driver should set the Enable Endian Word Swap bit. By setting this bit, the translation would be as follows:

Internal Byte Ordering

	31	1615		0
0x00	88	89	8A	8B
0x04	8C	8D	8E	8F

PCI Byte Ordering

31	1615		0	
88	89	8A	8B	0x00
8C	8D	8E	8F	0x04

Figure 93: Word Swap Enable Translation on 32-bit PCI (no byte swap)

The only side effect for a little endian host that sets the Enable Endian Word Swap bit would be that the driver would have to perform an additional word swap on any 64-bit fields (e.g., a 64-bit physical address) that were given to the driver by the Network Operating System (NOS).

Little-endian hosts will not want to set the Enable Endian Byte Swap bit for target accesses. This bit is intended to be used by big endian systems that needed PCI data (little endian) translated back to big endian format.

Note: Some big endian systems automatically already do this depending on the architecture of the hosts PCI to memory interface.

The following figures show the translation of data when the Enable Endian Byte Swap bit is set:

Internal Byte Ordering

	31	16	15	0
0x00	88	89	8A	8B
0x04	8C	8D	8E	8F

PCI Byte Ordering

31	16	15	0	
8F	8E	8D	8C	0x00
8B	8A	89	88	0x04

Figure 94: Byte Swap Enable Translation on 32-bit PCI (no word swap)

Internal Byte Ordering

	31	16	15	0
0x00	88	89	8A	8B
0x04	8C	8D	8E	8F

PCI Byte Ordering

31	16	15	0	
8B	8A	89	88	0x00
8F	8E	8D	8C	0x04

Figure 95: Byte and Word Swap Enable Translation on 32-bit PCI

WORD SWAP DATA AND BYTE SWAP DATA BITS

The Word Swap Data, and Byte Swap Data bits effect how packet data is ordered on the PCI bus. These only affect how packet data is ordered, and do not affect non-frame data (i.e., buffer descriptors, statistics block, etc.). In other words, these bits effect how data is transferred to/from host send/receive buffers.

For example, if BCM57XX family were to receive a packet that had the following byte order:

01	*02*	*03*	*04*	*05*	*06*	*07*	*08*	*09*	*0A*	*0B*	*0C*	*0D*	*0E*	*0F*	*10*
D1	D2	D3	D4	D5	D6	S1	S2	S3	S4	S5	S6	T1	T2	IP1	IP2

Where:

- D1-D6 consists of the packet’s destination address (Byte D0 is the first byte on the wire);
- S1-S6 is the source address;
- T1-T2 is the Ethernet type/length field;
- IP1-IP2 are the first two bytes of the IP header which immediately follow the type/length field.

The packet would be stored internally in big endian Format:

Table 101: Big-Endian Internal Packet Data Format

B0	*B1*	*B2*	*B3*	*B4*	*B5*	*B6*	*B7*
63-56	*55-48*	*47-40*	*39-32*	*31-24*	*23-16*	*15-8*	*7-0*
D1	D2	D3	D4	D5	D6	S1	S2
S3	S4	S5	S6	T1	T2	IP1	IP2

However, when the data gets transferred across PCI, there could be confusion about the correct byte ordering because PCI is Little Endian whereas BCM57XX family is a Big Endian device. So, in order to provide flexibility for different host processor/memory architectures, BCM57XX family can order this data on PCI in four different ways depending on the settings of the Word Swap Data, and Byte Swap Data bits. The below figures illustrate how data would appear on the PCI AD[63:0] pins depending on the settings of those swap bits:

Word Swap Data = 0, and Byte Swap Data = 0

Table 102: 64-bit PCI Bus (WSD = 0, BSD = 0)

B7	*7B6*	*B5*	*B4*	*B3*	*B2*	*B1*	*B0*
63-56	*55-48*	*47-40*	*39-32*	*31-24*	*23-16*	*15-8*	*7-0*
D1	D2	D3	D4	D5	D6	S1	S2
S3	S4	S5	S6	T1	T2	IP1	IP2

Table 103: 32-bit PCI Bus (WSD = 0, BSD = 0)

B3	*B2*	*B1*	*B0*
31-24	*23-16*	*15-8*	*7-0*
D5	D6	S1	S2
D1	D2	D3	D4
T1	T2	IP1	IP2
S3	S4	S5	S6

Word Swap Data = 0, and Byte Swap Data = 1

Table 104: 64-bit PCI Bus (WSD = 0, BSD = 1)

B7	*B6*	*B5*	*B4*	*B3*	*B2*	*B1*	*B0*
63-56	*55-48*	*47-40*	*39-32*	*31-24*	*23-16*	*15-8*	*7-0*
D4	D3	D2	D1	S2	S1	D6	D5
S6	S5	S4	S3	IP2	IP1	T2	T1

Table 105: 32-bit PCI Bus (WSD = 0, BSD = 1)

B3	*B2*	*B1*	*B0*
31-24	*23-16*	*15-8*	*7-0*
S2	S1	D6	D5
D4	D3	D2	D1
IP2	IP1	T2	T1
S6	S5	S4	S3

Word Swap Data = 1, and Byte Swap Data = 0

Table 106: 64-bit PCI Bus (WSD = 1, BSD = 0)

B7	B6	B5	B4	B3	B2	B1	B0
63-56	55-48	47-40	39-32	31-24	23-16	15-8	7-0
D5	D6	S1	S2	D1	D2	D3	D4
T1	T2	IP1	IP2	S3	S4	S5	S6

Table 107: 32-bit PCI Bus (WSD = 1, BSD = 0)

B3	B2	B1	B0
31-24	23-16	15-8	7-0
D1	D2	D3	D4
D5	D6	S1	S2
S3	S4	S5	S6
T1	T2	IP1	IP2

Word Swap Data = 1, and Byte Swap Data = 1

Table 108: 64-bit PCI Bus (WSD = 1, BSD = 1)

B7	B6	B5	B4	B3	B2	B1	B0
63-56	55-48	47-40	39-32	31-24	23-16	15-8	7-0
S2	S1	D6	D5	D4	D3	D2	D1
IP2	IP1	T2	T1	S6	S5	S4	S3

Table 109: 32-bit PCI Bus (WSD = 1, BSD = 1)

B3	B2	B1	B0
31-24	23-16	15-8	7-0
D4	D3	D2	D1
S2	S1	D6	D5
S6	S5	S4	S3
IP2	IP1	T2	T1

So, for a little-endian (e.g., x86) host, software should set both the Word Swap Data, and Byte Swap Data bits. This is because a little endian host will expect the first byte on the wire (byte D1) to be placed into memory at the least significant (starting) address of the packet data.

Word Swap Non-Frame Data and Byte Swap Non-Frame Data Bits

The Word Swap Non-Frame Data, and Byte Swap Non-Frame Data bits affect the byte ordering of certain shared memory data structures (buffer descriptors, statistics block, etc.) when those structures are transferred across PCI.

For example, lets examine how a Send Buffer Descriptor is stored internally in the BCM57XX family:

Table 110: Send Buffer Descriptor (Big-Endian 64-bit format)

Byte #	0	1	2	3	4	5	6	7	
Bit #	63	48	47	32	31	16	15	0	
	MSB			Host Address				LSB	0x00
	MSB Length	LSB	Flags		Reserved		VLAN		0x08

Since the BCM57XX family uses a 64-bit memory subsystem, the above diagram is shown in 64-bit format. Furthermore, the table shows both the internal byte offset for each field and the bit position for each byte.

Note: This may seem confusing because big-endian notation normally has the bit positions incrementing from left to right. However, in this case, the bit positions are relevant because they correspond to the bit positions on PCI (AD[63:0]) if neither of the non-frame data swap bits are set. For clarification, the following table shows the same structure in 32-bit format.

Table 111: Send Buffer Descriptor (Big-Endian 32-bit format)

Byte #	0	1	2	3	
Bit #	31	16	15	0	
	MSB	Host Address			0x00
				LSB	0x04
	MSB Length	LSB	Flags		0x08
	Reserved		VLAN		0x0C

In order to provide flexibility for different host processor/memory architectures, the BCM57XX family can order the data in memory in four different ways depending on the settings of the Word Swap Non-Frame Data and Byte Swap Non-Frame Data bits. The following tables show how data will appear depending on the settings of those swap bits:

Word Swap Non-Frame Data = 0 and Byte Swap Non-Frame Data = 0

This would require the software to use the following little-endian data structure on the host:

Table 112: Send Buffer Descriptor (Little-Endian 32-bit format) with No Swapping

Byte #	3	2	1	0	
Bit #	31	16	15	0	
		Host Address		LSB	0x00
	MSB				0x04
	Reserved		VLAN		0x08
	MSB Length LSB		Flags		0x0C

In this case, the data structure takes on a slightly new format because the words have been swapped.

Word Swap Non-Frame Data = 1 and Byte Swap Non-Frame Data = 0

This would require the software to use the following little-endian data structure on the host:

Table 113: Send Buffer Descriptor (Little-Endian 32-bit format) with Word Swapping

Byte #	3	2	1	0	
Bit #	31	16	15	0	
	MSB	Host Address			0x00
				LSB	0x04
	MSB Length LSB		Flags		0x08
	Reserved		VLAN		0x0C

The disadvantage of this approach is if the host operating system supported a 64-bit data type for a physical address, the host device driver would have to swap the two 32-bit words that comprise the 64-bit address that the host operating system used.

Word Swap Non-Frame Data = 0 and Byte Swap Non-Frame Data = 1

This would require the software to use the following big-endian data structure on the host:

Table 114: Send Buffer Descriptor (Big-Endian 32-bit format) with Byte Swapping

Byte #	0	1	2	3	
Bit #	31	16	15	0	
	MSB	Host Address			0x00
				LSB	0x04
	Reserved		VLAN		0x08
	MSB Length LSB		Flags		0x0C

Word Swap Non-Frame Data = 1 and Byte Swap Non-Frame Data = 1

This would require the software to use the following big-endian data structure on the host:

Table 115: Send Buffer Descriptor (Big-Endian 32-bit format) with Word and Byte Swapping

Byte #	0	1	2	3	
Bit #	31	16	15	0	
	MSB	Host Address			0x00
				LSB	0x04
	MSB	Length LSB	Flags		0x08
	Reserved		VLAN		0x0C

Section 10: Ethernet Link Configuration

OVERVIEW

The BCM57XX family supports multiple link operating modes. It can operate at multiple link speeds: 10 Mbps, 100 Mbps, or 1000 Mbps. It can also operate at half-duplex (IEE802.3 CSMA/CD) or full-duplex. The BCM57XX family can be configured to use MII, GMII, or TBI modes to interface with physical layer transceiver devices (a.k.a. PHYs) that support Ethernet networks that operate over either copper or fiber-optic wiring. The MAC is compliant with IEEE 802.3, 802.3u, 802.3x, and 802.3z specifications.

GMII/MII

The Gigabit Media Independent Interface (GMII) is normally used to interface the controller to a transceiver that supports Gigabit Ethernet over copper wiring (1000BASE-T). The Media Independent Interface (MII) is used to interface the controller to a transceiver that is capable of 10/100 Mbps Ethernet. NICs that support triple-speed operation (10/100/1000 Mbps) will have the BCM5700 MAC interface to a GMII/MII PHY (e.g., the Broadcom BCM5401 PHY) that is capable of triple-speed operation. The rest of the BCM57XX family includes both a MAC and PHY on-chip.

In order to operate a NIC running in GMII/MII mode, driver software will need to configure both the BCM57XX controller and, if the BCM5700 MAC is being used, the attached PHY as described below.

Note: The BCM57XX controllers with an integrated transceiver (or PHY) are hardwired with a PHY address of 1.

CONFIGURING THE BCM57XX FAMILY FOR GMII/MII AND TBI MODES

Configuring the BCM57XX family to operate in GMII or MII mode is simple. During initialization, software should configure the Ethernet_MAC_Mode.Port_Mode bits to a value that corresponds to the correct interface speed (01b for MII, 10b for GMII). The Ethernet_MAC_Mode.Port_Mode bits of BCM5700 should be configured for GMII/MII mode when it is connected to the external PHY through GMII/MII interface and for TBI mode when it is connected to the external SerDes PHY for 1000-BASE-X connections. The Ethernet_MAC_Mode.Port_Mode bits of BCM5701 should be configured to GMII/MII when it is using the on-chip triple speed Ethernet Transceiver and to TBI when it is using the TBI interface to talk to an external SerDes PHY. The Ethernet_MAC_Mode.Port_Mode bits of BCM5702, BCM5703C, BCM5704C, BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714C, BCM5714S, BCM5715C, and BCM5715S should be configured for GMII/MII as these devices interface with integrated on-chip PHY through GMII/MII. Note that the BCM5714S and BCM5715S MACs interface with the on-chip 1000-BASE-X PHY through the GMII interface. The Ethernet_MAC_Mode.Port_Mode bits of BCM5703S and BCM5704S devices should be configured to TBI as these devices interface with integrated on-chip SerDes PHY through TBI.

Configuring Link Up/Down

The BCM57XX family has two different methods that it can use to determine if the Ethernet link is up or down. The link will be down if the Ethernet cable is not properly attached at both ends of the network. Link will be up only if the cable is properly attached and the devices at both ends of the cable recognize that link has been established. The device cannot successfully transfer packets on the link unless it determines that it has a valid link up.

The first method for configuring link status is called auto-polling. Software can enable auto-polling by setting the MI_Mode.Port_Polling bit. If enabled, the BCM57XX family will periodically generate MDIO cycles to read the PHY's Link_Status bit in MII Status register (see "MI Status Register (Offset 0x450)" on page 389). The link status from the auto-polling operation is then reported in the MI_Status_Register.Link_Status and Transmit_MAC_Status.Link_Up (see "Transmit MAC Status Register (Offset 0x460)" on page 390) bits.

The second method for configuring link status involves using the BCM57XX family's LNKRDY input pin. This method allows the BCM57XX family to determine the link status based on an input from an external source (i.e., a link status output from a PHY). By default, the state of the LINKRDY pin sets the Transmit_MAC_Status.Link Up bit. The polarity of the LNKRDY signal can be programmed via the Ethernet_MAC_Mode.Link_Polarity bit.

Lastly, the link state of the BCM57XX family can also be forced by disabling both the auto-polling function and the LNKRDY signal and forcing the link status by directly writing to the MI_Status.Link_Status bit.

Link Status Change Indications

It is often desirable for host software to know when the status of the Ethernet link has changed. To generate an interrupt to the host when link status changes, software should set the Ethernet_MAC_Event_Enable.Link_State_Changed bit (see "Ethernet MAC Event Enable Register (Offset 0x408)" on page 381) and the Mode_Control.Interrupt_on_MAC_Attention bit (see "Mode Control Register (Offset 0x6800)" on page 502). With this configuration, the Ethernet_MAC_Status.Link_State_Changed bit and Link_State_Changed bit in the status block (see "Status Block" on page 103) will be set when the link has changed state.

Configuring the GMII/MII PHY

GMII/MII transceivers (PHYs) contain registers that a software driver can manipulate to change parameters in the PHY. These parameters include the link speed or duplex that the PHY is currently running at, or the speed/duplex options that the PHY advertises during the auto-negotiation process. NIC device drivers will typically access PHY registers during the driver initialization process in order to configure the PHYs speed/duplex or to examine the results of the auto-negotiation process (if enabled). For more information about PHY registers, refer to "Transceiver Registers" on page 603.

These PHY registers are accessed via a process called MDIO. The PHY will have two pins (MDIO and MDC) that connect the BCM5700 MAC to the PHY (e.g., BCM5401 PHY). Software accesses a PHY's registers via MDIO through the BCM5700's MI_Communication register (see "MI Communication Register (Offset 0x44C)" on page 388). By manipulating this register, software can read or write registers in a PHY. The BCM5700 accesses the external PHY through the external MDIO interface while the BCM5701 and later devices access the on-chip PHY through the internal MDIO interface. The device driver accesses the PHY registers through the MI_Communications registers of BCM57XX family and hence the driver software for accessing PHY is same in accessing either the external PHY of BCM5700 or internal on-chip PHY of BCM5701 and later devices. The following Pseudocode describes accessing the PHY registers through the MI_Communications registers of the BCM57XX NetXtreme family.

Reading a PHY Register

```
// If auto-polling is enabled, temporarily disable it
If (AutoPolling_Enabled == TRUE) Then
Begin
   Mi_Mode.PortPolling = 0
End

// Setup the value that we are going to write to MI Communication register
// Set bit 27 to indicate a PHY read.
// Set bit 29 to indicate the start of a MDIO transaction
Value32 = ((PhyAddress << 21) | (PhyRegOffset << 16) | 0x28000000)

// Write value to MI communication register
Mi_Communication_Register = Value32

// Now read back MI Communication register until the start bit
// has been cleared or we have timed out (>5000 reads)
Loopcount = 5000
While (LoopCount > 0)
Begin
   Value32 = Mi_Communication_Register
   If (!(Value32 | 0x20000000)) then BREAK loop
   Else Loopcount--
End

// Print message if error
If (Value32 | 0x20000000) then
Begin
   // It a debug case - cannot read PHY
   Procedure (Print Error Message)
   Value32 = 0
End

// If auto-polling is enabled, turn it back on
If (AutoPolling_Enabled == TRUE) then
Begin
   Mi_Mode.PortPolling = 1
End

// Now return the value that we read (lower 16 bits of reg)
Return (Value32 & 0xffff)
```

Writing a PHY Register

```
// If auto-polling is enabled, temporarily disable it
If (AutoPolling_Enabled == TRUE) Then
Begin
    Mi_Mode.PortPolling = 0
End

// Setup the value that we are going to write to MI Communication register
// Set bit 26 set to indicate a PHY write.
// Set bit 29 to indicate the start of a MDIO transaction
// The lower 16 bits equal the value we want to write to the PHY register
Value32 = ((PhyAddress << 21) | (PhyRegOffset << 16) | RegValue | 0x24000000)

// Write value to MI communication register
Mi_Communication_Register = Value32

// Now read back MI Communication register until the start bit
// has been cleared or we have timed out (>5000 reads)
Loopcount = 5000
While (LoopCount > 0)
Begin
    Value32 = Mi_Communication_Register
    If (!(Value32 | 0x20000000)) then BREAK loop
    Else Loopcount--
End

// Print message if error
If (Value32 | 0x20000000) then
Begin
    // It a debug case - can't write PHY
    Procedure (Print Error Message)
    Value32 = 0
End

// If auto-polling is enabled, turn it back on
If (AutoPolling_Enabled == TRUE) Then
Begin
    Mi_Mode.PortPolling = 1
End
```

TBI MODE

This mode is only applicable to the BCM5700, BCM5701, BCM5703S, and BCM5704S devices.

The Ten-Bit Interface (TBI) is used to interface the controller to a SerDes device that provides transceiver function for a 1000BASE-SX connection. 1000BASE-SX is the most commonly used standard for Gigabit Ethernet functionality over networks that use fiber-optic cabling. NICs that use the TBI interface run only at gigabit speeds. Triple speed operation (10/100/1000 Mbps) is not supported in TBI mode.

However, auto-negotiation can still be performed in TBI mode. Although the NIC only supports a single speed in TBI mode, the level of flow control supported along with link status can be successfully negotiated with the link partner. Unlike GMII/MII PHYs though, SerDes devices do not normally support the IEEE 802.3z auto-negotiation protocol, and thus this function should be performed by the controller in concert with device driver software.

CONFIGURING THE BCM57XX FAMILY FOR TBI MODE

To configure the BCM57XX family to operate in TBI mode, software must configure the Ethernet_MAC_Mode.Port_Mode bits (see “Ethernet MAC Mode Register (Offset 0x400)” on page 377) to the value 11b during initialization.

1000BASE-X AUTO-NEGOTIATION

The BCM57XX family provides the ability for the host software to implement 1000BASE-X auto-negotiation function as described in IEEE 802.3z. The basic mechanism to achieve auto-negotiation is to exchange configuration data with the link partner through the Transmit_Gigabit_Auto_Negotiation (at offset 0x444) and Receive_Gigabit_Auto_Negotiation (at offset 0x448) registers. These registers are only valid when the BCM57XX family is operating in TBI mode (see “TBI Block” on page 72).

The host software sends configuration data by first initializing the Transmit_Gigabit_Auto_Negotiation with the data and then setting the Ethernet_MAC_Mode.Send_Configs bit. The BCM57XX family will continuously send out the configuration data, with the most significant bit first, until the Send_Configs bit is cleared.

The Receive_Gigabit_Auto_Negotiation register holds the received configuration data. When the BCM57XX family is receiving configuration data, the Ethernet_MAC_Status.Receiving_Config bit is set. If new configuration data is received, the Ethernet_MAC_Status.Config_Changed bit is set. The host software will have to clear this bit after reading the new configuration data.

Figure 96 shows how the auto-negotiation configuration word is encoded into the Receive_Gigabit_Auto_Negotiation and Transmit_Gigabit_Auto_Negotiation registers.

MSB															LSB
D7	D6	D5	D4	D3	D2	D1	D0	D15	D14	D13	D12	D11	D10	D9	D8

Figure 96: Auto-Negotiation Configuration Word Encoding

1000BASE-X Auto-Negotiation in Firmware

The BCM5701 and BCM5703S controllers support firmware auto-negotiation for the TBI interface. The BCM5704S supports auto-negotiation in hardware or firmware.

- When the system begins in the cold boot state, firmware always tries to do auto-negotiation with Link Partner and set a result into the shared memory area between the host driver and firmware.
- In warm boot (Device Reset), the auto-negotiation can be done in either the driver or firmware.

The integrated PHY of BCM5714C, BCM5714S, BCM5715C, and BCM5715S devices supports the 1000BASE-X auto-negotiation in hardware.

Communication between the host driver and firmware is done through a dedicated shared memory location.

Shared Memory Mailbox, Signatures, and Bit Definitions

```
// Local memory address for the FW_TBI_MAILBOX
#define FW_TBI_MAILBOX 0xC1C

// Lower 16 bits are valid for driver with this signature in the upper 16 bits
#define FW_TBI_SIG                                                           0x736b0000
// 1 = Link is up and auto-negotiation was successful
#define FW_TBI_LINK_OK BIT_0

// 1 = Link partner advertised symmetric pause capability
#define FW_TBI_AUTONEG_LP_SYM_PAUSE                                          BIT_1
// 1 = Link partner advertised asymmetric pause capability
#define FW_TBI_AUTONEG_LP_ASYM_PAUSE                                         BIT_2
// 1 = Fallback to 1000FD
#define FW_TBI_AUTONEG_FALLBACKBIT_3

// Lower 16 bits are valid for firmware with this signature in the upper 16 bits
#define DRV_TBI_SIG                                                          0x686b0000
// 1 = Enable auto-negotiation in firmware
#define DRV_TBI_ENABLE_AUTONEG                                               BIT_0
// 1 = Advertise symmetric pause capability for auto-negotiation
#define DRV_TBI_ADV_SYM_PAUSE                                                BIT_1
// 1 = Advertise asymmetric pause capability for auto-negotiation
#define DRV_TBI_ADV_ASYM_PAUSE                                               BIT_2
// 1 = Fallback to 100FD if auto-negotiation fails
#define DRV_TBI_AUTONEG_FALLBACKBIT_3
```

If the link partner is not sending out configuration data for flow control during auto-negotiation, the auto-negotiation firmware tries to set the link to 1000 full-duplex (HD) mode with the link partner.

Firmware function during reset:

- From cold boot, the firmware always performs auto-negotiation and writes the FW_TBI_SIG value, along with the results of auto-negotiation, to the FW_TBI_MAILBOX shared memory location.
- From warm boot (Device Reset), the firmware checks for DRV_TBI_SIG in the FW_TBI_MAILBOX. If this signature is not present, the firmware does not perform the auto-negotiation function and does not configure the TBI interface. This is necessary to allow legacy driver to operate normally.
- From warm boot, if the DRV_TBI_SIG value is present, the firmware only attempts auto-negotiation if the DRV_TBI_ENABLE_AUTONEG bit is set in the FW_TBI_MAILBOX shared memory location. Otherwise, the firmware does not attempt auto-negotiation and the driver is expected to perform auto-negotiation.

Host driver function:

- During host driver initialization, the driver checks for FW_TBI_SIG at the mailbox location FW_TBI_MAILBOX in the shared memory area. If the signature is not present or Link status is not successful, the driver performs the auto-negotiation function and TBI configuration. If the signature is present, the auto-negotiation function and TBI configuration are done by firmware.
- After device reset with the Miscellaneous Configuration register (see "Miscellaneous Configuration Register (Offset 0x6804)" on page 504) Core Clock Blocks Reset bit and before setting the T3_MAGIC_NUM (i.e., KevT) into shared memory location 0x0b50, the driver places DRV_TBI_SIG and auto-negotiation parameters in the mailbox location FW_TBI_MAILBOX.
- The host driver periodically polls the FW_TBI_MAILBOX for the auto-negotiation result from the firmware in the mailbox location.

MDI REGISTER ACCESS

Configuring physical devices (such as the BCM5401 PHY, and so on) and querying the status of physical devices are done via the MDIO interface (MDC and MDIO).

Note: This procedure is PHY-independent. The MAC access to the PHY is the same for the entire BCM57XX family.

There are four modes in which the internal MII Management interface signals (MDC/MDIO) can be controlled for communication with the internal transceiver registers. These modes are as follows:

- Mode 1: Autopolling Mode. Enabled by setting the Enable bit in the MAC Ethernet MI Mode register (see "MI Mode Register (Offset 0x454)" on page 389). The device will poll for the link status bit in the transceiver.
- Mode 2: Command Control. Writing to the MI Communications register (see "MI Communication Register (Offset 0x44C)" on page 388) directly to either read or write the transceiver registers.
- Mode 3: Management Interface Program. By setting the Enable_MIP bit in the MAC Mode register (see "Ethernet MAC Mode Register (Offset 0x400)" on page 377) and programming the start/end memory area in the MBUF1 memory space that contain the register read/write commands. The chip will then transfer these commands one by one into the MI Communications register to preform the actual read/write operations.
- Mode 4: Register Control. When enabled, the host writes the MDC clock and data values into the MDI Control register (see "MDI Control Register (Offset 0x6844)" on page 515). The values in the register directly control the interface signals.

Mode 1 has the lowest priority and it will be stalled any time there is an active operation through the MI Communications register (mode 2). Modes 3 and 4 have the highest priority. When mode 3 or 4 is enabled, the MI Communications register cannot be read or written. Mode 4 is enabled by setting the MDI Select bit of the MDI Control register.

OPERATIONAL CHARACTERISTICS

The interface between the MAC and physical devices is with two signals: MDIO clock (MDC) and bidirectional Serial Data (MDIO). The details of the MDIO interface can be found in the physical device data sheet or IEEE 802.3 specification.

ACCESS METHODS

The MAC provides two methods to access the Physical Device registers via the MDIO interface:

- Traditional bit-bang method
- Auto-access method

Traditional Bit-Bang Method

In this method, software has to toggle the MDC and MDIO pins to access physical device registers. Software is responsible for providing the proper delay to satisfy timing requirements such as setup time, hold time, clock frequency, etc.

The MDI_Control_Register (see "MDI Control Register (Offset 0x6844)" on page 515) controls the MDC and MDIO pins as follows:

- MDI_Control_Register.MDI_Clock: is used to clock MDC pin. If this bit is set, MDC is logic high. Otherwise, it is logic low.
- MDI_Control_Register.MDI_Select: This bit is used to select either the traditional bit-bang method or auto-access method. This bit must be set to 1 when the traditional bit-bang method is used.
- MDI_Control_Register.MDI_Enable: This bit is used to control bi-directional data pin. If this bit is set to 1, then MDI_Control_Register.MDI_Data is output. Otherwise, it is input.
- MDI_Control_Register.MDI_Data: This is used to control bi-directional MDIO pin.

Auto-access Method

The BCM57XX device has a built-in interface to access physical device registers without having to control MDC and MDIO pins by software/firmware. It provides an easy way to access the physical device register.

To use this mode, MDI_Control_Register.MDI_Select has to be cleared to 0. The MI_Communication_Register (see "MI Communication Register (Offset 0x44C)" on page 388) is used to access physical device.

Note: Programmers must be careful to wait for the start _busy bit to clear. Writing to the MI Communication register (see "MI Communication Register (Offset 0x44C)" on page 388) prior to the completion of a previous MDI access will yield unpredictable MDI data. The previous access will not complete successfully.

For example, to read a 16-bit physical register at offset 0x2 from a physical device which is strapped to PHY address 1, perform the following steps:

1. MI_Communication_Register.Register_Address is set to 0x2.
2. MI_Communication_Register.PHY_Addr is set to 1.
3. MI_Communication_Register.Command is set to 0x2.
4. MI_Communication_Register.Start_Busy is set to 1.
5. Poll Until MI_Communication_Register.Start_Busy is cleared to 0.
6. MI_Communication_Register.Transaction_Data contains 16-bit physical register.

Refer to "Configuring the GMII/MII PHY" on page 242 for example code.

To write a 16-bit physical register at offset 0x3 from a physical device which is strapped to PHY address 1, perform the following steps:

1. MI_Communication_Register.Register_Address is set to 0x3.
2. MI_Communication_Register.PHY_Addr is set to 1.
3. MI_Communication_Register.Command is set to 0x1.
4. MI_Communication_Register.Start_Busy is set to 1.
5. Poll Until MI_Communication_Register.Start_Busy is cleared to 0.

Refer to "Configuring the GMII/MII PHY" on page 242 for example code.

PHY Setup and Initialization

Physical layer link changes may be reflected to the MAC through the following mechanisms:

- Physical Layer Interrupts
- Autopolling
- Link Ready Signal from PHY

There are two methods of determining link change status from device driver:

- PIO read to the Ethernet MAC status register 0x404 (see "Ethernet MAC Status Register (Offset 0x404)" on page 379).
- Read the Link_State_Changed bit in the status block (see "Status Block" on page 103).

Note: Refer to the application schematics for a detailed understanding of the PHY and MAC connections. Physical Layer Interrupts and Link Ready signals may be routed differently on PCB layouts. Host software has dependency on the hardware implementation. Programmers are strongly encouraged to reference the schematics for their board level application.

BCM5700 MAC and BCM5401 PHY (Example)

Host software may configure the physical layer to assert a PHY interrupt when link state changes. The hardware design will have a trace routed between the PHY $\overline{\text{INTR}}$ and MAC $\overline{\text{MDINT}}$ pins (see Figure 97). When the PHY detects a change in link state, an interrupt signal is asserted. Host software must configure the PHY's interrupt mask register so the appropriate event can be reflected/asserted. Additionally, host software must configure the MAC to reflect the assertion as a host line driven interrupt ($\overline{\text{INTA}}$).

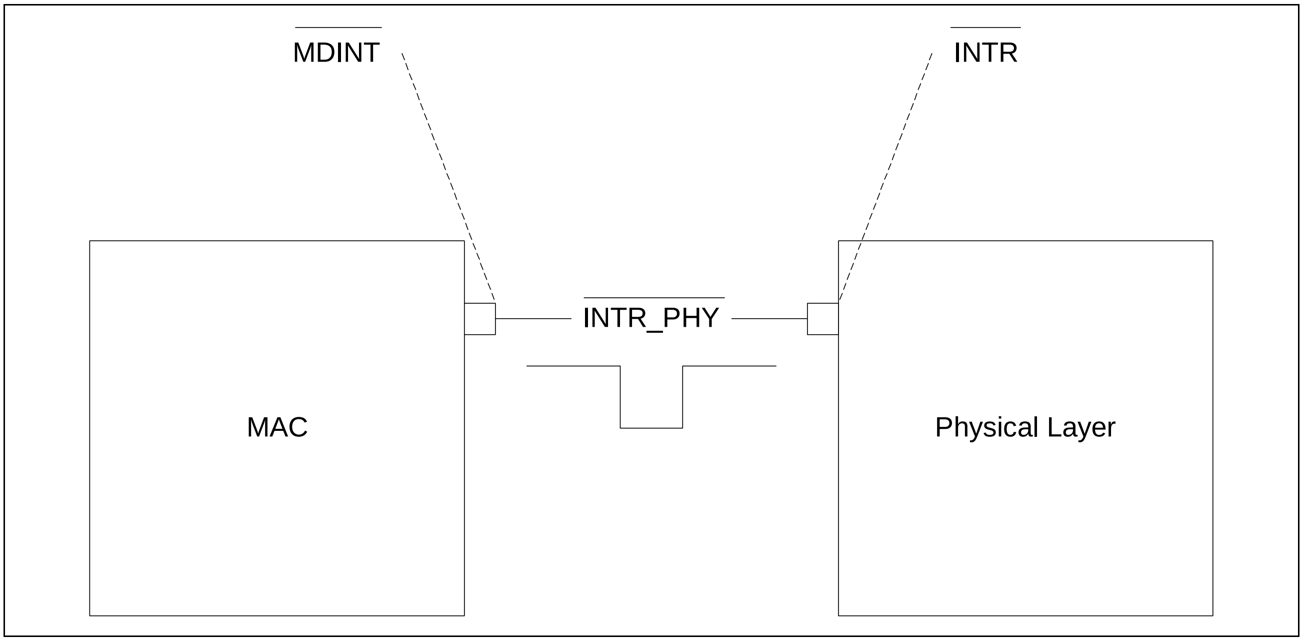


Figure 97: Trace Routed Between PHY $\overline{\text{INTR}}$ and MAC $\overline{\text{MDINT}}$ Pins

The BCM5700 MAC can poll the PHY's MDI status register 0x02 at physical layer address 0x01 (see Figure 98). This mechanism is called auto-polling, since the MAC will periodically check the MDI status register for changes in link state. When the MDI Link Status bit changes, the MAC will assert an attention to the host using the INTA signal.

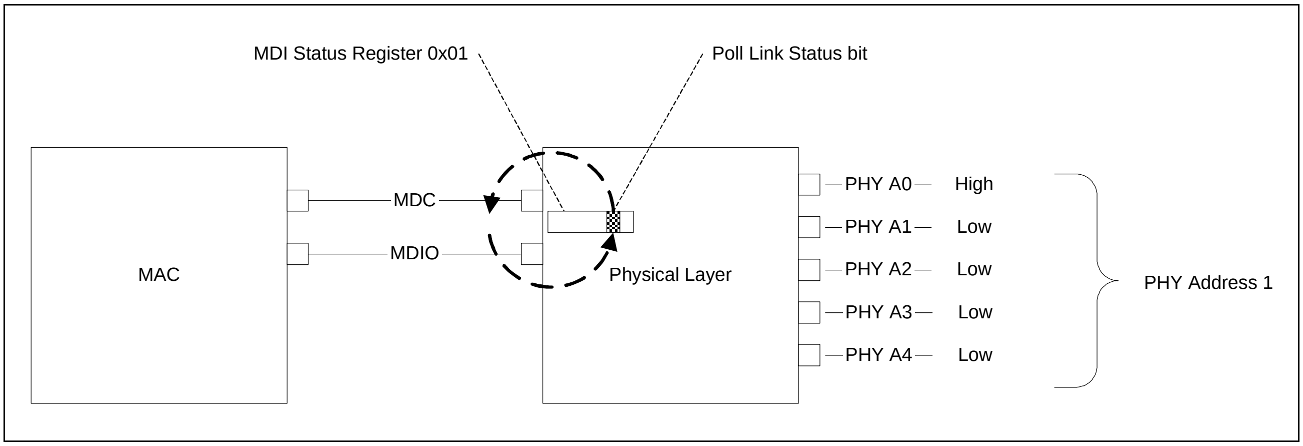


Figure 98: Polling the PHY's MDI Status Register

For example, the Link Ready signal may be routed from the BCM5401 Link10 pin to the LNKRDY pin on the MAC (see Figure 99). The BCM5401 Link10 pin is active low and the MAC LNKRDY pin is active high. Host software must disable the BCM5401 Three Link LED Mode setting, so the Link10 signal indicates link up for all wire speeds (10/100/1000 Mbps). The Link Ready mode is mutually exclusive with PHY interrupt mode, and the autopolling mode must be disabled. Software must toggle polarity of the MAC's LNKRDY pin, since the pin is active high—reset default.

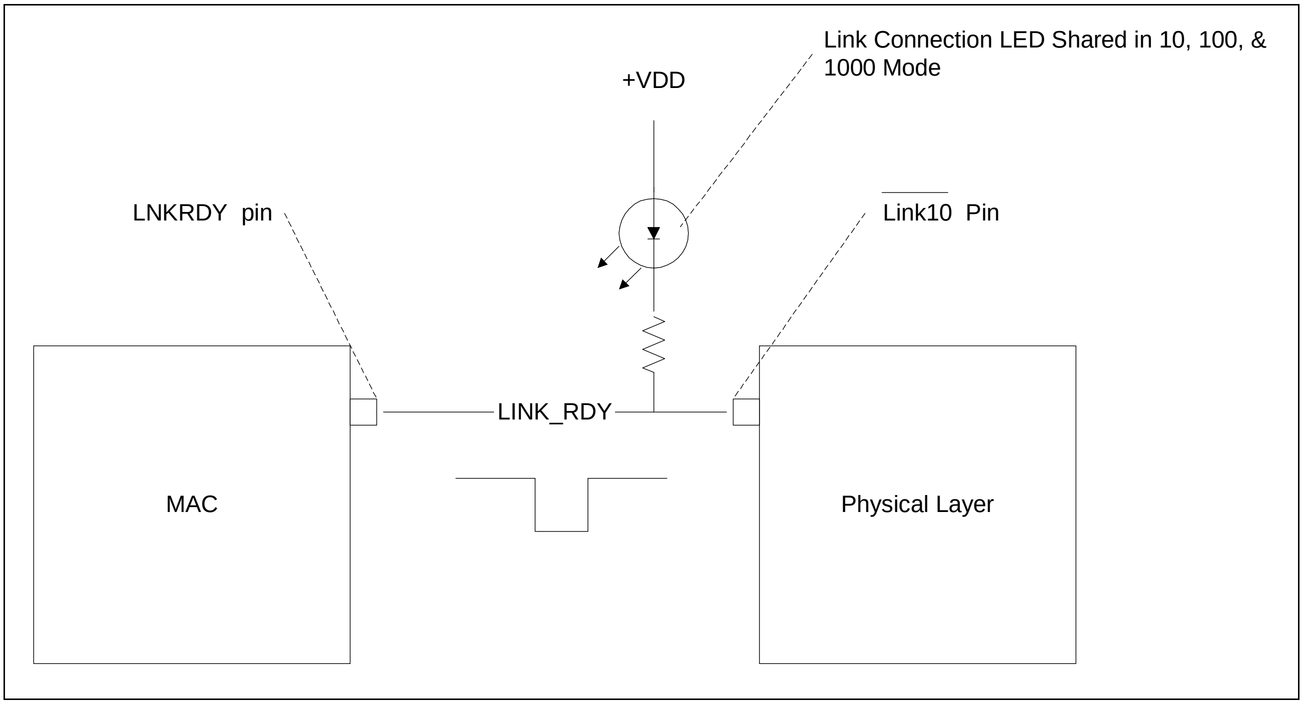


Figure 99: Routing the Link Ready Signal

The BCM5700 MAC's LNKRDY pin can be programmed for active high or active low logic assertion (see Figure 100). An active high signal is typically associated with a high voltage and an active low signal to a low voltage. Board layouts may route different signals to the LNKRDY pin, and software must configure the logic level appropriately. Software may set the logic level of the BCM5700 LNKRDY signal with the Link_Polarity bit in the Ethernet MAC Mode register (see "Ethernet MAC Mode Register (Offset 0x400)" on page 377).

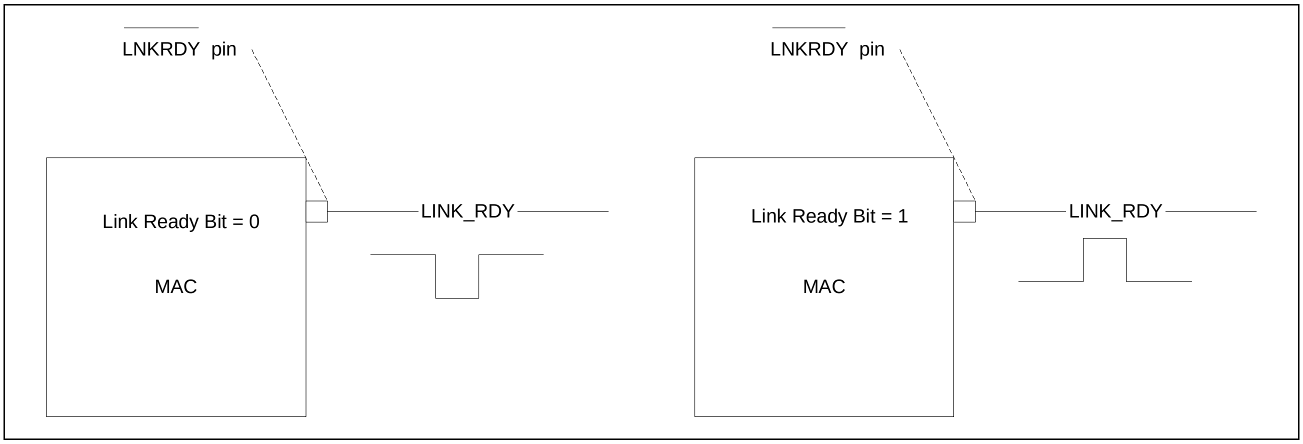


Figure 100: LNKRDY Pin Programming (Active High or Active Low)

Setup and Initialization Procedure

This section lists the initializing procedure for the BCM5401 PHY.

1. Disable link events. Write the value 0x00 to Ethernet MAC Event Enable register (see "Ethernet MAC Event Enable Register (Offset 0x408)" on page 381). This step de-asserts events for the MI_Interrupt and Link_State_Changed.
2. Clear link attentions. Write the Sync_Changed and Config_Changed bits in the Ethernet MAC Status register (see "Ethernet MAC Status Register (Offset 0x404)" on page 379) to clear pending attentions.
3. Disable Autopolling mode. Write the value 0xC000 to the MI Mode register (see "MI Mode Register (Offset 0x454)" on page 389). The Port_Polling bit will be de-asserted.
4. Wait 40 µs for the Auto Poll disable step to complete.
5. Disable BCM5401 WOL mode. Write 0x02 to the PHY's MDI Auxiliary Control register. This value will enable the Power Control Shadow register, disable WOL, and configure normal operation.
6. The programmer is strongly encouraged to reference the errata sheet for the physical layer component. Implement the appropriate workarounds.
7. Acknowledge outstanding interrupts by reading the MDI Interrupt_Status register twice. There are sticky bits, which require host software to read the register twice, to clear values.
8. Configure the PHY interrupt mask. Clear the Link_Status_Change bit in the MDI Interrupt_Mask register. Each bit in the MDI Interrupt_Status bit has a 1:1 mapping to bits in the MDI Interrupt_Mask register. Assert all remaining bits to disable/mask out other interrupt types.
9. Configure the Three LINK LED Mode bit (Optional). The configuration of this bit is dependent on the board layout. This bit should be asserted when PHY interrupts are enabled. This bit should be cleared under the following conditions:
   - BCM5401 Link10 (or similar signal) is routed to LNKRDY on MAC
   - PHY interrupts are disabled
   - Autopolling is disabled.

10. Determine link status. Read the MDI Status register twice. There are sticky bits that must be cleared—second read clears bits latched high.
11. Determine speed and duplex operation. Read the MDI Auxiliary_Status register. The programmer may poll the MDI Auxiliary_Status register for 20 ms and check the Link_Status pass bit.
12. Store speed/duplex settings in driver state variables. Initialization complete if no link detected.
13. Enable Flow Control. Reference “Flow Control” on page 280.
14. Configure MAC port mode. The driver state variables from step 12 are used to configure the Port_Mode bit field in the Ethernet MAC Mode register (see “Ethernet MAC Mode Register (Offset 0x400)” on page 377). This step should be completed each time link changes—the port mode does not reflect the type of PHY attached to the MAC, rather Port_Mode configures the hardware interface between MAC and PHY.
15. Configure duplex mode. The driver state variables from step 12 are used to configure the Half_Duplex bit in the Ethernet MAC Mode register (see “Ethernet MAC Mode Register (Offset 0x400)” on page 377).
16. Configure link polarity (Optional). Software may configure the BCM5700 to indicate link status changes through the LNKRDY pin. The polarity of the signal routed from the PHY may be toggled using the Link_Polarity bit in the Ethernet MAC Mode register (see “Ethernet MAC Mode Register (Offset 0x400)” on page 377).
17. Enable port polling (Optional). Software may set the Port_Polling bit in the MI_Mode register (see “MI Mode Register (Offset 0x454)” on page 389).
18. Enable link attentions. Software should enable link attentions, so a host line interrupt is driven when link state changes. Programmers may choose the type of attentions generated with the Ethernet MAC Event Enable register (see “Ethernet MAC Event Enable Register (Offset 0x408)” on page 381):
    - PHY interrupt attentions are enabled by asserting the MI_Interrupt bit.
    - Port polling attentions are enabled by asserting the Link_State_Changed bit.
    - LNKRDY attentions are also enabled by asserting the Link_State_Changed bit.

The BCM5464 PHY core, used in BCM5751, BCM5721, and BCM5705, shows poor BER performance when the Ethernet connection is linked at gigabit speed with a cable length of 70m of less. This is caused by the internal hybrid BIAS current level, which impacts the slew-rate of the transmitted signals. To work around this issue, the firmware and/or driver need to perform the following writes to the PHY module immediately after the PHY comes out of reset. The firmware v3.19a or later has this work around implemented.

1. Write the value 0x0C00 to PHY register at offset 0x18
2. Write the value 0x000A to PHY register at offset 0x17
3. Write the value 0x310B to PHY register at offset 0x15
4. Write the value 0x201F to PHY register at offset 0x17
5. Write the value 0x9506 to PHY register at offset 0x15
6. Write the value 0x401F to PHY register at offset 0x17
7. Write the value 0x14E2 to PHY register at offset 0x15
8. Write the value 0x0400 to PHY register at offset 0x18

Pseudocode to Set Up Fiber Auto-Negotiation

The auto-negotiation-related attributes (e.g., mr_an_complete, mr_an_enable) are inherited from the 2000 edition IEEE STD 802.3 figure 37.6—auto-negotiation state diagram.

Hardware register read/write always uses an assignment expression with an identifier postfixed with _register. For example:

- Register read:
  `Value32_variable = MAC_status_register`
- Register write:
  `MAC_mode_register.send_configs = 1`
  `transmit_1000Base-X_AN_register = 0`

Register reference documentation should be used to understand various register bit field manipulation.

Any identifiers without ending _register are software representation. They do not change any hardware registers.

```
/* function forward declaration */
AN_result_return_value Autoneg8023z(Input_parameter BCM570x_data.AN_Information)
FC_return_value Setup_Flow_Control(BCM570x_SW_representation   BCM570x_data,
   input_unsigned_32bit_param LocalPhyAd_variable,
   input_unsigned_32bit_param RemotePhyAd_variable)
return_value Indicate_Status(BCM570x_SW_representation BCM570x_data,
   input_parameter Status)

Setup_Fiber_Phy(BCM570x_SW_representation BCM570x_data)
{
   local_unsigned_32bit_variable     CurrentLinkStatus_variable ;
   local_unsigned_32bit_variable     Value32_variable ;
   local_unsigned_32bit_variable     Count_variable ;
   local_unsigned_32bit_variable     j_variable ;
   local_unsigned_32bit_variable     MacMode ;
   local_enum_variable                    AnStatus_variable ;

   /* Enable full-duplex mode in MAC mode register (offset 0x400) */
   MAC_mode_register.Half_Duplex = 0 ;

   /* Enable TBI mode */
   MAC_mode_register.Port_Mode = TBI_mode ;

   /* Fill Transmit 1000Base-X AN register (offset 0x444) with zero */
   Transmit_1000Base-X_AN_register = 0 ;

   if (phy_device == BCM8002_SerDes_phy)
   {
      initialize_BCM8002 ;
   }

   /* Set link status change interrupt bit in MAC event enable register (offset 0x408) */
   MAC_event_enable_register.Link_State_Changed = 1 ;

   /* Assume current link status to be down */
   CurrentLinkStatus_variable = link_down ;

   /* Read the link status out of MAC status register (offset 0x404) */
   Value32_variable = MAC_status_register ;
```

```
if ( ( PCS_synched_bit in Value32_variable ) = 1 ) /* if link is up */
{
   /* If auto-negotiation(AN) mode is intended, which should be default */
   if (BCM570x_data.requested_media_type is media_type_AN)
   {
      /* Initialize the AN default capabilities. */
      BCM570x_data.AN_infomation.full_duplex = enabled ;
      BCM570x_data.AN_infomation.symmetrical_pause = enabled ;
      BCM570x_data.AN_infomation.asymmetrical_pause = enabled ;
      BCM570x_data.AN_infomation.autoneg = enabled ;

      /* Get intended flow_control setting */
      Value32_variable = BCM570x_data.flow_control_setting ;
      /* bit 10 is the pause capable bit */
      if (Value32_variable.bit_10 is set)
         BCM570x_data.AN_infomation.pause_capable = 1 ;
      else
         BCM570x_data.AN_infomation.pause_capable = 0 ;

      /* bit 11 is the asymmetric pause bit */
      if (Value32_variable.bit_11 is set)
         BCM570x_data.AN_infomation.asymmetric_pause = 1 ;
      else
         BCM570x_data.AN_infomation.asymmetric_pause = 0 ;

      /* Try to auto-negotiate up to four times. */
      for_loop (Count_variable from 1 to 4)
      {
         /* Fill transmit 1000Base-X AN register with zero */
         transmit_1000Base-X_AN_register = 0 ;

         /* Write zero to the GRC timer register (offset 0x680C) */
         GRC_timer_register = 0 ;

         /* Initialize autoneg state to unknown state */
         BCM570x_data.AN_information.State = AN_ unknown_state ;

         /* Initialize current time to zero u-sec */
         BCM570x_data.AN_information.CurrentTime_us = 0 ;
         while_loop( BCM570x_data.AN_information.CurrentTime_us < 55000 )
         {
            /* Call function Autoneg8023z to do IEEE802.3z auto negotiation */
            AnStatus_variable = Autoneg8023z( BCM570x_data.AN_information ) ;

            if ( AnStatus_variable is AN_done or AN_failed )
               break_out_of_while_loop ;

            /* Update timer variable with hardware timer value */
            BCM570x_data.AN_information.CurrentTime_us = GRC_timer_register ;
         }  /* End of while_loop */

         /* Auto-negotiation finishes normally */
         if ( AnStatus_variable is AN_done )
            break_out_of_for_loop ;

         /* Auto-negotiation finishes abnormally (cable may be disconnected) */
```

```
             if ( AnStatus_variable is AN_failed )
                break_out_of_for_loop ;

             /* Break out the retry loop early, if there is no cable. */
             if ( Count_variable >= 1 )
             {
                /* Read MAC status register (offset 0x404) into Value32_variable */
                Value32_variable = MAC_status_register ;
                /* If PCS_synced bit (bit 0) in Value32_variable is 0, break out */
                if ( Value32_variable.PCS_synced is 0 )
                   break_out_of_for_loop ;
             }
          } /* End of for_loop */

          /* Write 0 to MAC_mode_register.send_configs bit to stop sending configs */
          MAC_mode_register.send_configs = 0 ;

          /* Resolve follow control settings. */
          if ( (AnStatus_variable is AN_done ) AND
             ( BCM570x_data.AN_infomation.mr_an_complete bit is set ) AND
             ( BCM570x_data.AN_infomation.mr_link_ok bit is set ) AND
             ( BCM570x_data.AN_infomation.mr_lp_adv_full_duplex bit is set ) )
          {
             local_unsigned_32bit_variable   LocalPhyAd_variable ;
             local_unsigned_32bit_variable   RemotePhyAd_variable ;

             /* initialize local variable */
             LocalPhyAd_variable = 0 ;
             RemotePhyAd_variable = 0 ;

             if ( BCM570x_data.AN_infomation.mr_adv_sym_pause bit is set )
                LocalPhyAd_variable.bit_10 = 1 ;

             if ( BCM570x_data.AN_infomation.mr_adv_asym_pause bit is set )
                LocalPhyAd_variable.bit_11 = 1 ;

             if ( BCM570x_data.AN_infomation.mr_lp_adv_sym_pause bit is set )
                RemotePhyAd_variable.bit_10 = 1 ;

             if ( BCM570x_data.AN_infomation.mr_lp_adv_asym_pause bit is set )
                RemotePhyAd_variable_bit_11 = 1 ;

             /* Call flow control function to set up flow control registers */
             Setup_Flow_Control( BCM57XX_data, LocalPhyAd_variable, RemotePhyAd_variable
);

             CurrentLinkStatus_variable = link_active ;
          }

          /* Clear the interrupt status. */
          /* Set bit 3 (Config changed) and bit 4 (Sync Changed) of */
          /* MAC status register (OFFSET 0X404) to 1       */
          MAC_status_register.bit_3 = 1 ;
          MAC_status_register.bit_4 = 1 ;

          /* Make sure we really acknowledge the interrupt, otherwise we */
```

```
      /* will enter this routine again. To ensure that we can       */
      /* successfully clear the interrupt, we need to wait for the   */
      /* other side stop sending configs. This case will only occur */
      /* when we are connected back-to-back.                         */
      if ( AnStatus_variable == AN_done )
      {
         /* wait for link partner to finish */
         for_loop ( when j_variable is from 1 to 400 )
         {
            wait_100_u-second ;
            /* Read MAC status register (offset 0X404) into Value32_variable */
            Value32_variable = MAC_status_register ;
            /* Continue waiting until bit 2 (Receiving config) is cleared */
            if ( Value32_variable.bit_2 is cleared )
            break_out_of_for_loop; /* The interrupt will be cleared below */
         }
      }
   }
   else /* End of if(BCM570x_data.requested_media_type is media_type_AN) */
   {
      /* We are forcing line speed 1000FD */
      CurrentLinkStatus_variable = link_active ;
   }
} /* End of if( (PCS_synched_bit in Value32_variable) = 1 ) */

/* Write 0 to Ethernet MAC Mode Register (offset 0x400) bit 10 */
/* to set the link polarity bit to positive polarity           */
MAC_mode_register.bit_10 = 0 ;

/* Clear the interrupt status again. */
/* Write a 1 to bit 3 (Config changed) and bit 4 (Sync Changed)      */
/* to Ethernet MAC Status Register (offset 0x404) to clear interrupt */
MAC_status_register.bit_3 = 1 ;
MAC_status_register.bit_4 = 1 ;
/* Update software flags to indicate link status change */
BCM570x_data.virtual_block.status.bit_0 = 1 ; /* Status is now updated */
BCM570x_data.virtual_block.status.bit_1 = 0 ; /* Link status change cleared */

/* Initialize the current link status. */
if ( CurrentLinkStatus_variable == link_active )
{
   /* Link is up, set software flagsto indicate it */
   BCM570x_data.line_speed = line_speed_1000Mbps ;
   BCM570x_data.duplex_mode = duplex_mode_full ;
}
else
{
   /* Link is down, set unknown state */
   BCM570x_data.line_speed = line_speed_unknown ;
   BCM570x_data.duplex_mode = duplex_mode_unknown ;
}

/* Indicate media type and link status. */
BCM570x_data.media_type = media_type_fiber ;
BCM570x_data.link_status = CurrentLinkStatus_variable ;
/* call Indicate_Status to notify OS that fiber phy has been set up */
```

```
   Indicate_Status(BCM570x_data, CurrentLinkStatus_variable) ;

   return Success;
}

/* This function implements IEEE 802.3z Figure 37-6 Auto-Negotiation state diagram */
/* Reference IEEE 802.3 Section 37 to get complete comprehension */

AN_result_return_value Autoneg8023z(Input_parameter BCM570x_data.AN_Information)
{
   local_unsigned_16bit_variable     RXConfig_variable ;
   local_unsigned_32bit_variable     Delta_us_variable ;
   local_enum_variable                AN_Result_variable ; /* AN result/return variable */

   /* Initialize all AN flags to default values */
   if ( BCM570x_data.AN_Information.State is AN_unknown_state )
   {
      BCM570x_data.AN_Information.RXConfig = 0 ;
      BCM570x_data.AN_Information.CurrentTime_us = 0 ;
      BCM570x_data.AN_Information.LinkTime_us = 0 ;
      BCM570x_data.AN_Information.AbilityMatchCfg = 0 ;
      BCM570x_data.AN_Information.AbilityMatchCnt = 0 ;
      BCM570x_data.AN_Information.AbilityMatch = AN_fase ;
      BCM570x_data.AN_Information.IdleMatch = AN_fase  ;
      BCM570x_data.AN_Information.AckMatch = AN_fase  ;
   }

   /* Set the AbilityMatch, IdleMatch, and AckMatch flags if their */
   /* corresponding conditions are satisfied. */
   /* Read receive_1000Base-X_AN_register (OFFSET 0X448) to a local variable */
   if ( Reading receive_1000Base-X_AN_register to RXConfig_variable succeeds )
   {
      if ( RXConfig_variable != BCM570x_data.AN_Information.AbilityMatchCfg )
      {
         BCM570x_data.AN_Information.AbilityMatchCfg = RXConfig_variable ;
         BCM570x_data.AN_Information.AbilityMatch = AN_fase ;
         BCM570x_data.AN_Information.AbilityMatchCnt = 0 ;
      }
      else
      {
         BCM570x_data.AN_Information.AbilityMatchCnt++ ;
         if ( BCM570x_data.AN_Information.AbilityMatchCnt > 1 )
         {
            BCM570x_data.AN_Information.AbilityMatch = AN_true ;
            BCM570x_data.AN_Information.AbilityMatchCfg = RXConfig_variable ;
         }
      }

      if ( RXConfig_variable.AN_config_ack_bit is set )
         BCM570x_data.AN_Information.AckMatch = AN_true ;
      else
         BCM570x_data.AN_Information.AckMatch = AN_fase ;

         BCM570x_data.AN_Information.IdleMatch = AN_fase ;
   }
   else  /* Reading receive_1000Base-X_AN_register fails */
```

```
{
   BCM570x_data.AN_Information.IdleMatch = AN_true ;
   BCM570x_data.AN_Information.AbilityMatchCfg = 0 ;
   BCM570x_data.AN_Information.AbilityMatchCnt = 0 ;
   BCM570x_data.AN_Information.AbilityMatch = AN_fase ;
   BCM570x_data.AN_Information.AckMatch = AN_fase ;

   RXConfig_variable = 0 ;
}

/* Save the last RX_Config value */
BCM570x_data.AN_Information.RXConfig = RXConfig_variable ;

/* Default return code. */
AN_Result_variable = AN_status_ok ;

/* AN state machine as defined in 802.3z/802.3 section 37.3.1.5. or Fig. 37-6 */
switch( BCM570x_data.AN_Information.State )
{
   case AN_unknown_state:
      /* Check management register(mr) AN flags */
      if(BCM570x_data.AN_Information.mr_an_enable or
         BCM570x_data.AN_Information.mr_restart_an)
      {
         BCM570x_data.AN_Information.CurrentTime_us = 0 ;
         BCM570x_data.AN_Information.State = AN_enable_state ;
      }

      /* Fall through */

   case AN_enable_state:
      BCM570x_data.AN_Information.mr_an_complete = AN_fase ;
      BCM570x_data.AN_Information.mr_page_rx = AN_fase ;

      if ( BCM570x_data.AN_Information.mr_an_enable )
      {
         BCM570x_data.AN_Information.LinkTime_us = 0 ;
         BCM570x_data.AN_Information.AbilityMatchCfg = 0 ;
         BCM570x_data.AN_Information.AbilityMatchCnt = 0 ;
         BCM570x_data.AN_Information.AbilityMatch = AN_fase ;
         BCM570x_data.AN_Information.IdleMatch = AN_fase ;
         BCM570x_data.AN_Information.AckMatch = AN_fase ;

         BCM570x_data.AN_Information.State = AN_restart_init ;
      }
      else
         BCM570x_data.AN_Information.State = AN_disable_link_ok ;

      break_out_of_switch ;

   case AN_restart_init:
      BCM570x_data.AN_Information.LinkTime_us =
      BCM570x_data.AN_Information.CurrentTime_us ;
      BCM570x_data.AN_Information.mr_np_loaded = AN_fase ;

      /* Write 0 to a buffer (BCM570x_data.AN_Information.TXConfig) */
      BCM570x_data.AN_Information.TXConfig = 0;
   /* Copy buffer to transmit 1000Base-X AN register (offset 0x444) */
   transmit_1000Base-X_AN_register = BCM570x_data.AN_Information.TXConfig ;
   /* Write 1 to MAC_mode_register.send_configs bit to send data in TX register */
   MAC_mode_register.send_configs = 1 ;

   AN_Result_variable = AN_timer_enabled ;
   BCM570x_data.AN_Information.State = AN_restart ;

   /* Fall through.*/

case AN_restart:
   /* Get the current time and compute the delta with the saved link timer */
   Delta_us_variable = BCM570x_data.AN_Information.CurrentTime_us -
          BCM570x_data.AN_Information.LinkTime_us ;
   if ( Delta_us_variable > 9000 )
      BCM570x_data.AN_Information.State = AN_ability_detect_init ;
   else
      AN_Result_variable = AN_timer_enabled ;

   break_out_of_switch ;

case AN_disable_link_ok:
   AN_Result_variable = AN_done;

   break_out_of_switch ;

case AN_ability_detect_init:
   /* Note: in the state diagram, this variable is set to */
   /* mr_adv_ability<12>. Is this right?. */
   BCM570x_data.AN_Information.mr_toggle_tx = AN_fase;

   /* The TXConfig register is arranged as follows:                      */
   /*                                                                    */
   /* MSB                                                           LSB  */
   /* +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+  */
   /* | D7| D6| D5| D4| D3| D2| D1| D0|D15|D14|D13|D12|D11|D10| D9| D8|  */
   /* +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+  */
   /* Fill buffer (BCM570x_data.AN_Information.TXConfig) with */
   /* desired data(advertisement) */
   BCM570x_data.AN_Information.TXConfig = 0; /* initialize buffer */
   BCM570x_data.AN_Information.TXConfig.D5 =
   BCM570x_data.AN_Information.mr_adv_full_duplex ;
   BCM570x_data.AN_Information.TXConfig.D6 =
   BCM570x_data.AN_Information.mr_adv_half_duplex ;
   BCM570x_data.AN_Information.TXConfig.D7 =
   BCM570x_data.AN_Information.mr_adv_sym_pause ;
   BCM570x_data.AN_Information.TXConfig.D8 =
   BCM570x_data.AN_Information.mr_adv_asym_pause ;
   BCM570x_data.AN_Information.TXConfig.D12 =
   BCM570x_data.AN_Information.mr_adv_remote_fault1 ;
   BCM570x_data.AN_Information.TXConfig.D13 =
   BCM570x_data.AN_Information.mr_adv_remote_fault2 ;
   BCM570x_data.AN_Information.TXConfig.D15 =
   BCM570x_data.AN_Information.mr_adv_next_page ;
```

```
         /* Copy buffer to transmit 1000Base-X AN register (offset 0x444) */
         transmit_1000Base-X_AN_register = BCM570x_data.AN_Information.TXConfig ;
         /* Write 1 to MAC_mode_register.send_configs bit to send data in TX register */
         /* Send the config as advertised in the advertisement register */
         MAC_mode_register.send_configs = 1 ;

         BCM570x_data.AN_Information.State = AN_ability_detect ;

         break_out_of_switch ;

      case AN_ability_detect:
         if ( BCM570x_data.AN_Information.AbilityMatch == AN_true AND
               BCM570x_data.AN_Information.RXConfig != 0 )
            BCM570x_data.AN_Information.State = AN_ack_detect_init ;

         break_out_of_switch ;

      case AN_ack_detect_init:
         /* Write 1 to a buffer (BCM570x_data.AN_Information.TXConfig.Bit14_ACK) */
         BCM570x_data.AN_Information.TXConfig.Bit14_ACK = 1;
         /* Copy buffer to transmit 1000Base-X AN register (offset 0x444) */
         transmit_1000Base-X_AN_register = BCM570x_data.AN_Information.TXConfig ;
         /* Write 1 to MAC_mode_register.send_configs bit to send data in TX register */
         MAC_mode_register.send_configs = 1 ;

         BCM570x_data.AN_Information.State = AN_ack_detect ;

         /* Fall through. */

      case AN_ ack_detect:
         if ( BCM570x_data.AN_Information.AckMatch == AN_true )
         {
            if ( ( BCM570x_data.AN_Information.RXConfig & 0xFFBF ) ==
               ( BCM570x_data.AN_Information.AbilityMatchCfg & 0xFFBF ) )
               BCM570x_data.AN_Information.State = AN_complete_ack_init ;
            else
               BCM570x_data.AN_Information.State = AN_enable_state ;
         }
         else if ( BCM570x_data.AN_Information.AbilityMatch == AN_true AND
               BCM570x_data.AN_Information.RXConfig == 0 )
         {
            BCM570x_data.AN_Information.State = AN_enable_state;
         }

         break_out_of_switch;

      case AN_complete_ack_init:
         /* The RXConfig register is arranged as follows:                     */
         /*                                                                   */
         /* MSB                                                          LSB  */
         /* +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+  */
         /* | D7| D6| D5| D4| D3| D2| D1| D0|D15|D14|D13|D12|D11|D10| D9| D8|  */
         /* +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+  */

         /* Make sure invalid bits are not set. */
         if ( BCM570x_data.AN_Information.RXConfig.D0  OR
         BCM570x_data.AN_Information.RXConfig.D1  OR
         BCM570x_data.AN_Information.RXConfig.D2  OR
         BCM570x_data.AN_Information.RXConfig.D3  OR
         BCM570x_data.AN_Information.RXConfig.D4  OR
         BCM570x_data.AN_Information.RXConfig.D9  OR
         BCM570x_data.AN_Information.RXConfig.D10 OR
         BCM570x_data.AN_Information.RXConfig.D11 )
      {
         AN_Result_variable = AN_failed ;
         break_out_of_switch;
      }

      /* Set up the link partner advertisement register. */
      BCM570x_data.AN_Information.mr_lp_adv_full_duplex =
      BCM570x_data.AN_Information.RXConfig.D5 ;
      BCM570x_data.AN_Information.mr_lp_adv_half_duplex =
      BCM570x_data.AN_Information.RXConfig.D6 ;
      BCM570x_data.AN_Information.mr_lp_adv_sym_pause =
      BCM570x_data.AN_Information.RXConfig.D7 ;
      BCM570x_data.AN_Information.mr_lp_adv_asym_pause =
      BCM570x_data.AN_Information.RXConfig.D8 ;
      BCM570x_data.AN_Information.mr_lp_adv_remote_fault1 =
      BCM570x_data.AN_Information.RXConfig.D12 ;
      BCM570x_data.AN_Information.mr_lp_adv_remote_fault2 =
      BCM570x_data.AN_Information.RXConfig.D13 ;
      BCM570x_data.AN_Information.mr_lp_adv_next_page =
      BCM570x_data.AN_Information.RXConfig.D15 ;

      BCM570x_data.AN_Information.LinkTime_us =
      BCM570x_data.AN_Information.CurrentTime_us ;

      BCM570x_data.AN_Information.mr_toggle_tx =
      BCM570x_data.AN_Information.mr_toggle_tx ;
      BCM570x_data.AN_Information.mr_toggle_rx =
      BCM570x_data.AN_Information.RXConfig.D11 ;
      BCM570x_data.AN_Information.mr_np_rx =
      BCM570x_data.AN_Information.RXConfig.D15 ;
      BCM570x_data.AN_Information.mr_page_rx = AN_true ;

      BCM570x_data.AN_Information.State = AN_complete_ack ;
      AN_Result_variable = AN_timer_enabled ;

      break_out_of_switch ;

   case AN_complete_ack:
      if ( BCM570x_data.AN_Information.AbilityMatch == AN_true AND
         BCM570x_data.AN_Information.RXConfig == 0 )
      {
         BCM570x_data.AN_Information.State = AN_enable_state ;
      break_out_of_switch ;
      }

      Delta_us_variable = BCM570x_data.AN_Information.CurrentTime_us -
         BCM570x_data.AN_Information.LinkTime_us ;

      if ( Delta_us_variable > 9000 )
   {
      if ( BCM570x_data.AN_Information.mr_adv_next_page == 0 OR
         BCM570x_data.AN_Information.mr_lp_adv_next_page == 0 )
      {
         BCM570x_data.AN_Information.State = AN_idle_detect_init ;
      }
      else
      {
         if ( BCM570x_data.AN_Information.TXConfig.D15 == 0 AND
            BCM570x_data.AN_Information.mr_np_rx == 0 )
            BCM570x_data.AN_Information.State = AN_idle_detect_init;
         else
            AN_Result_variable = AN_failed ;
      }
   }

   break_out_of_switch;

case AN_idle_detect_init:
   BCM570x_data.AN_Information.LinkTime_us =
   BCM570x_data.AN_Information.CurrentTime_us;

   /* Write 0 to MAC_mode_register.send_configs bit to stop sending configs */
   MAC_mode_register.send_configs = 0 ;

   BCM570x_data.AN_Information.State = AN_idle_detect ;

   AN_Result_variable = AN_timer_enabled ;

   break_out_of_switch ;

case AN_idle_detect:
   if ( BCM570x_data.AN_Information.AbilityMatch == AN_true AND
      BCM570x_data.AN_Information.RXConfig == 0 )
   {
      BCM570x_data.AN_Information.State = AN_enable_state ;
      break_out_of_switch ;
   }

   Delta_us_variable = BCM570x_data.AN_Information.CurrentTime_us -
      BCM570x_data.AN_Information.LinkTime_us ;
   if ( Delta_us_variable > 9000 )
      BCM570x_data.AN_Information.State = AN_link_ok ;

   break_out_of_switch ;

case AN_link_ok:
   BCM570x_data.AN_Information.mr_an_complete = AN_true ;
   BCM570x_data.AN_Information.mr_link_ok = AN_true ;
   AN_Result_variable = AN_done ;

   break_out_of_switch ;

case AN_next_page_wait_init:
   /* IEEE 802.3 NEXT_PAGE_WAIT state is optional and not implemented */
```

```
          break_out_of_switch;

      case AN_ next_page_wait:
          /* IEEE 802.3 NEXT_PAGE_WAIT state is optional and not implemented */

          break_out_of_switch;

          default:
          /* Invalid case, execution should never get here */
          AN_Result_variable = AN_failed ;
          break_out_of_switch;
   }

   return AN_Result_variable;
}


/* Acronyms for following functions:              */
/* FC = Flow Control, RX = Receive, TX = Transmit, */
/* AN = Auto-Negotiation, lp = link partner        */

FC_return_value Setup_Flow_Control(
   BCM570x_SW_representation   BCM570x_data,
   input_unsigned_32bit_param  LocalPhyAd_variable,
   input_unsigned_32bit_param  RemotePhyAd_variable)
{
   local_unsigned_32bit_variable flow_cap_variable ;

   flow_cap_variable = 0 ; /* initialization flow capability */

   /* Resolve flow control attributes */
   /* See Table 28B-3 of 802.3ab-1999 spec. */
   /* MSb(bit 31) of BCM570x_data.FlowControlCap indicates */
   /* if flow control auto pause mode turns on */
   if ( bit_31 of BCM570x_data.FlowControlCap is set )
   {
      /* Bit 10 of LocalPhyAd_variable indicates local phy auto- */
      /* negotiation(AN) advertisement register pause capability */
      if ( LocalPhyAd_variable.bit_10 is set )
      {
         /* LocalPhyAd_variable bit 11 indicates phy AN asym pause */
         if ( LocalPhyAd_variable.bit_11 is set )
         {
            /* RemotePhyAd_variable bit 10 indicates if AN remote */
            /* link partner pause capable turns on or not         */
            if ( RemotePhyAd_variable.bit_10 is set )
            {
               flow_cap_variable.bit_1 = 1 ; /* turn on FC RX pause */
               flow_cap_variable.bit_2 = 1 ; /* turn on FC TX Pause */
            }
            /* bit 11 sets AN remote link partner asymmetric pause */
            else if ( RemotePhyAd_variable.bit_11 is set )
            {
               flow_cap_variable.bit_1 = 1 ; /* turn on FC RX pause */
            }
         }
         else /* Local AN asymmetric pause mode turns off */
         {
            /* if Remote link partner pause capable turns on */
            if ( RemotePhyAd_variable.bit_10 is set )
            {
               flow_cap_variable.bit_1 = 1 ; /* turn on FC RX pause */
               flow_cap_variable.bit_2 = 1 ; /* turn on FC TX Pause */
            }
         }
      }
      /* Local phy AN advertisement asymmetric pause is turned on */
      else if ( LocalPhyAd_variable.bit_11 is set )
      {
         /* Both pause capable and asymmetric pause modes of AN */
         /* remote link partner ability register are turned on */
         if ( RemotePhyAd_variable.bit_10 AND bit_11 are both set )
            flow_cap_variable.bit_2 = 1 ; /* turn on FC TX Pause */
      }
   }
   else /* if auto pause mode is turned off, none of FC mode is set */
      flow_cap_variable = BCM570x_data.FlowControlCap ;

   /* Enable/disable RX PAUSE. */
   /* Bit 2 of Receive MAC Mode Register (offset 0x468) determines */
   /* flow control on or off. First, turn it off.                  */
    BCM570x_data.RXMode.bit_2 = 0 ;
   if ( ( flow_cap_variable.bit_1 is set) AND /* flow control RX pause is on */
      ( BCM570x_data.FlowControlCap.bit_31 is set OR /* FC auto pause is on */
         BCM570x_data.FlowControlCap.bit_1 is set ) ) /* FC RX pause is on */
   {
      BCM570x_data.FlowControl.bit_1 = 1 ; /* Turn on FC RX pause */
      BCM570x_data.RXMode.bit_2 = 1 ; /* Turn on flow control mode */
   }
   /* Write the data in BCM570x_data.RXMode into */
   /* Receive MAC Mode Register (offset 0x468) */
   receive_MAC_mode_register = BCM570x_data.RXMode ;

   /* Enable/disable TX PAUSE. */
   /* Bit 4 of Transmit MAC Mode Register (offset 0x45C) determines */
   /* flow control on or off. First, turn it off.                  */
   BCM570x_data.TXMode.bit_4 = 0 ;
   if ( ( flow_cap_variable.bit_2 is set) AND /* flow control TX pause is on */
      ( BCM570x_data.FlowControlCap.bit_31 is set OR /* FC auto pause is on */
         BCM570x_data.FlowControlCap.bit_2 is set ) ) /* FC TX pause is on */
   {
      BCM570x_data.FlowControli.bit_2 = 1 ;
      BCM570x_data.TXMode.bit_4 = 1 ;
   }
   /* Write the data in BCM570x_data.RXMode into */
   /* Transmit MAC Mode Register (offset 0x45C) */
   transmit_MAC_mode_register = BCM570x_data.TXMode ;

   return FC_success ;
}
```

```
return_value Indicate_Status(
   BCM570x_SW_representation BCM570x_data,
   input_parameter Status)
{
   local_boolean_variable TimerCancelled_variable;

   if ( BCM570x_data.requested_media_type is MAC_loopback OR phy_loopback )
   {
      BCM570x_data.link_status = link_active ;

      /* OS should not transmit packets if the link is down */
      if ( BCM570x_data.initialization_done )
      {
         /* If OS link and status are not updated yet, do it now */
         if ( BCM570x_data.OS_link_indication_status != BCM570x_data.link_status )
         {
            /* Different OS's handle resource access synchronization in */
            /* different way, follow your OS protocol to implement phy  */
            /* setup completion notification here. Generally, there is  */
            /* a resource-take and resource-release pair for guarding.  */
            /* Before entering here, OS should lock resource already.   */

            OS_specific_semaphore_release(input_parameters, ...) ;

            /* Notify OS of the current status - phy setup done    */
            OS_specific_indicate_status(BCM570x_data.connected, BCM570x_data.handler);

            /* OS should lock the resource again                   */
            OS_specific_semaphore_take(input_parameters, ...) ;
         }
      }

      BCM570x_data.OS_link_indication_status = link_active ;

      return success ;
   }

   if ( BCM570x_data.drivr_state != normal_mode )
      return success ;

   if ( BCM570x_data.OS_link_indication_status == BCM570x_data.link_status )
   {
      /* When the cable is plugged back, we may not finish auto-negotiation */
      /* successfully due to signal stability.  In this case we force      */
      /* an auto-negotiation retry 1 second later.                          */
      if ( ( BCM570x_data.requested_media_type == media_type_AN )  AND
         ( BCM570x_data.enable_TBI is true )  AND
         ( BCM570x_data.link_status == link_down )  AND
         ( BCM570x_data.initialization_done is true ) )
      {
         OS_specific_cancel_current_timer(input_parameters, ...) ;
         OS_specific_set_retry_after_1_sec(input_parameters, ...) ;
      }

      if ( ( BCM570x_data.link_status == link_down ) OR
         ( BCM570x_data.link_status == link_active AND
      BCM570x_data.OS_line_speed_indication == BCM570x_data.line_speed ) )
            return success ;
   }

   BCM570x_data.autoneg_retry_count = 0;

   if ( BCM570x_data.initialization_done )
   {
      /* Reset timer interval to fire a periodical signal */
      OS_specific_cancel_current_timer(input_parameters, ...) ;

      if ( BCM570x_data.link_status == link_active )
         OS_specific_set_timer_at_1.2_sec(input_parameters, ...) ;
      else
         OS_specific_set_timer_at_0.02_sec(input_parameters, ...) ;
   }
   else
   {
      BCM570x_data.OS_link_indication_status = BCM570x_data.device.link_status;
      BCM570x_data.OS_line_speed_indication = BCM570x_data.device.line_speed;
   }

   return success ;
}
```

WAKE-ON LAN MODE/LOW-POWER

DESCRIPTION

The BCM57XX family uses the ACPI D3 hot/cold (low-power) state to conserve energy. The OS power management policy notifies device drivers to initiate power management transitions. The BCM57XX device driver should move the MAC into the D3 hot/cold power state—a response to the power management request (see Appendix B ”PC Power Management” on page 708 for more details on power management). While the BCM57XX family is in a D3 state, the RX MAC will filter incoming packets. The RX MAC compares incoming traffic for Interesting Packet pattern matches. The BCM57XX family asserts the PCI $\overline{\text{PME}}$ signal, when a positive WOL packet comparison is made. The $\overline{\text{PME}}$ signal notifies the Operating System and host device driver to transition the MAC into the D0 (high power) state.

WOL mode is a combination of PHY and MAC configurations. Both the PHY and MAC must be configured correctly to enable Broadcom's WOL technology.

- The BCM5700 and BCM5701 evaluation boards only support D3 cold Wake-up at 10 Mbps. The BCM5701 chip consumes roughly 325 mA and the board level design must conserve the remaining 50 mA. The PCI power management specification does not allow devices to draw more than 375 mA in D3 states.
- While in the D3 hot state, the BCM5401 PHY (BCM5700 MAC only) and BCM57XX family can support wake-up while running at any line speed.

The BCM57XX family supports both Interesting Packet pattern matching the AMD Magic Packet proprietary technology for WOL. The WOL support for the AMD Magic Packet format does not require host software to configure a pattern filter. The Magic Packet comparison is made in hardware and is enabled through a register interface. The AMD Magic Packet can be either broadcast or directed, and must contain the receiver's MAC address at least six times (repeating) in the packet. The Magic Packet wake-up is configured different from pattern match wake-up.

The following components are involved in WOL operation:

- Internal memory
- WOL Pattern Pointer register
- WOL Pattern Configuration register
- WOL Streams
- Pattern Data Structure
- GPIO
- Firmware mailbox
- PHY Auto-negotiation
- PHY Power Management
- BCM57XX Power Management

FUNCTIONAL OVERVIEW

The BCM57XX family is capable of WOL in 10/100 Mbps for copper-based controllers and 1000 Mbps for fiber-based controllers.

Note: When configured for WOL in 1000-Mbps mode, the BCM57XX family draws more than the 375 mA allowed by the PCI specification.

The BCM5700 evaluation NIC and BCM5701 physical layer supports WOL at 10 Mbps wire speed so the host programmer should configure the MAC accordingly.

The BCM57XX family uses the TX FIFO to store pattern data (see Figure 101). During WOL operation, the transmit engine is disabled and its FIFO is free for use. The TDE fetches data from the memory arbiter starting at a location specified in the WOL_Pattern_Pointer register. The WOL pattern checker pulls data off the TX FIFO for packet comparisons. The RX MAC will move incoming frame(s) to the pattern checker, and the remaining RX data path is not utilized. A state machine controls the Magic Packet comparisons. The WOL state machine will move out of an Idle state, when ACPI power management is enabled. The WOL state machine will clear the TX FIFO and Match register. The Match register indicates a positive Magic Packet comparison(s) on a stream.

In 10/100 Mbit mode, data is received once every four clock cycles. The pattern checker compares the first three patterns in the first cycle, the second three patterns in the second cycle, and the third three patterns in the last cycle. It is idle during the fourth cycle. In gigabit mode, the pattern checker gets three pattern words from the FIFO at one time.

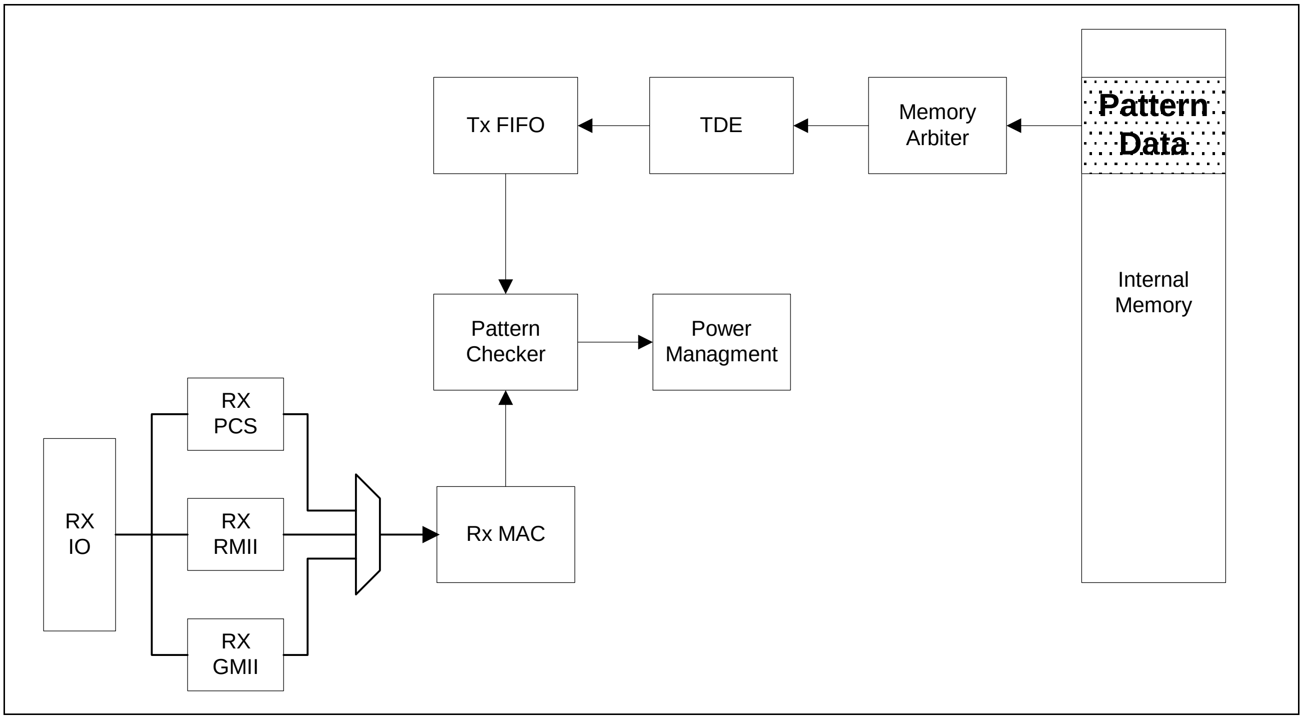


Figure 101: WOL Functional Block Diagram

Operational Characteristics

Internal Memory

The WOL pattern must be stored in the BCM57XX miscellaneous memory region. All memory locations require the host software to reinitialize the WOL pattern before each D0 to D3 transition. The RX/TX MAC places packets into this internal memory and the WOL pattern is overwritten during normal operation. When the BCM57XX family operates in D0 state, internal data structures use the same memory location as the WOL pattern. Host software should re-initialize the WOL pattern before each WOL sleep transition.

Table 116 shows the required memory regions for the WOL pattern.

Table 116: Required Memory Regions for WOL Pattern

Internal Address Range	*Size*	*Name*	*Cross-Reference*
0x8000–0x8FFF	8 KB	Miscellaneous Memory Region	Table 82 on page 171

WOL Pattern Pointer Register

The WOL_Pattern_Pointer specifies a location within BCM57XX address space where the pattern buffers reside (see "WOL Pattern Pointer Register (Offset 0x430)" on page 385 for the register definition). The internal memory subsection discusses how host programmers can choose an address range. The WOL_Pattern_Pointer register uses a pointer value, not an internal memory location. The pointer value is calculated by dividing an internal memory location by the value 8. Do not program the WOL_Pattern_Pointer register with the actual internal memory location. Rather, host software must first convert the base address to a pointer value. Here are example conversion from memory base to pointer values:

- 0x0000 (Misc Memory)/8 = 0x00 (required value)
- 0x400 (base addr)/8 = 0x80 (pointer value)
- 0x8000 (base addr)/8 = 0x1000 (pointer value)
- 0xF000 (base addr)/8 = 0x1E00 (pointer value)

WOL Pattern Configuration Register

The WOL_Pattern_Configuration register contains two programmable data fields. Both fields use different units of measurement, so the host programmer should be careful (see "WOL Pattern Configuration Register (Offset 0x434)" on page 386 for the register definition). This register is used to position and extract data from RX Ethernet frames.

- **Offset Field**—The Offset field in the WOL_Pattern_Configuration register specifies a position in RX Ethernet frame(s), where comparisons for WOL patterns should begin. This register uses a unit of measurement specified in terms of 2-byte chunks. Software should not program this field with a byte value, but should first normalize to a 2-byte unit. Hardware cannot begin WOL comparisons on odd byte alignments (i.e., 3,5,7,9 offsets). Host software must begin all pattern matching on even byte boundaries (i.e., 2,4,6,8 offsets). The 2 bytes per unit forces even byte alignment. For example:
    - 0x14 (byte offset)/2 = 0x0A (register ready)
    - 0x28 (byte offset)/2 = 0x14 (register ready)
    - 0xFC (byte offset)/2 = 0x7E (register ready)

- **Length Field**—The Length field in the WOL_Pattern_Configuration register specifies the number of clock cycles required to compare a variable number of bytes, in the RX stream. The Length field uses a unit of measurement specified in terms of memory arbiter clock cycles. Software should not program this field with a byte value. The Length field should be programmed with the maximum number of clocks required to compare the largest pattern size for the nine streams (10/100 mode only).

  Note: The BCM57XX family only supports one pattern stream at gigabit wire speed, so the length field will always be the largest pattern size.

  The programmer must use the following equation to calculate the number of clock cycles required to match patterns at 10/100 wire-speed: (Length/2) * 3 MA clocks. The equation breaks down as follows:

  - Determine the number of bytes in the RX Ethernet frame to compare. This value is a byte length.
  - The WOL pattern checker can compare two bytes simultaneously. Divide length by two bytes and round up to nearest integer value.
  - The BCM57XX family compares 2 bytes every three memory arbiter (MA) clock cycles. Multiply (Length/2) by three clock cycles.
- The following are example clock cycle calculations:
  - Data stream length = 25 bytes
  - 25 bytes/2 = 12.5 byte-pairs
  - Round(12.5) = 13 byte-pairs
  - 13 byte-pairs * 3 clocks/byte-pairs = 39 clocks (register ready)
  - Data stream length = 83 bytes
  - 83 bytes/2 = 41.5 byte-pairs
  - Round(41.5) = 42 byte-pairs
  - 42 byte-pairs * 3 clocks/byte-pair = 126 clocks (register ready)

WOL Streams

A stream is a comparison operation on RX frame(s). When the MAC is running at 10/100 Mbps wire speed, nine different patterns can be compared against the RX frame(s). The BCM57XX family moves RX frame(s) into nine parallel comparators, and the frame is matched simultaneously. The MAC is capable of filtering nine different patterns in 10/100 modes. The WOL pattern checker breaks frames into 2-byte pairs, so all nine comparators can begin matching data. In Figure 102, three Ethernet frames are compared against the nine available patterns.

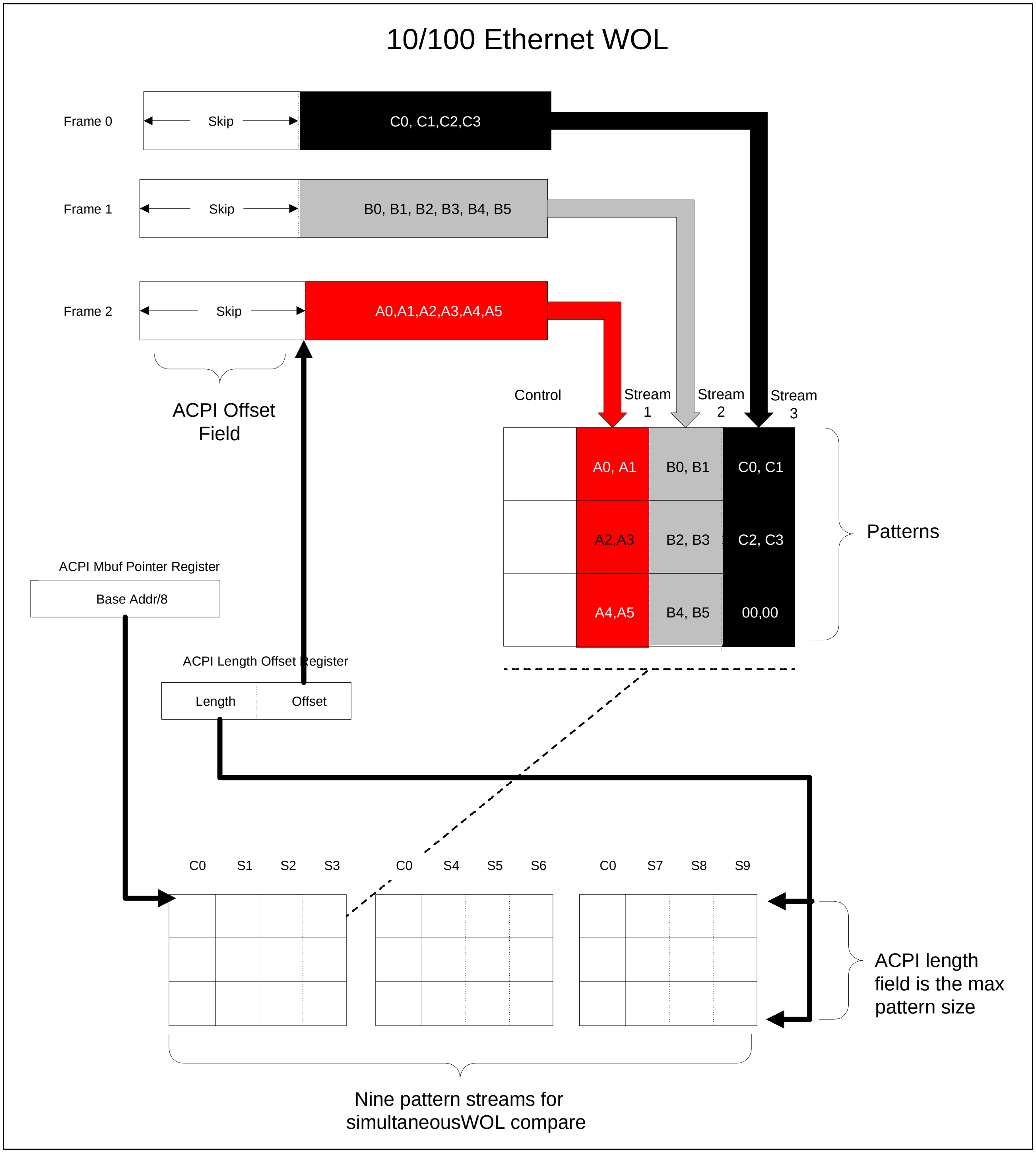


Figure 102: Comparing Ethernet Frames Against Available Patterns (10/100 Ethernet WOL)

Pattern Data Structure

The maximum number of entries in either 10/100 or 1000 mode is 128. The BCM57XX family cannot process a pattern that requires more than 128 entries. The size of an entry will vary based on 10/100 or 1000 Mbps mode. Additionally, all unused rows must be initialized with zeros. The WOL hardware cannot process an entry unless unused rows and rules have been zeroed out (see Figure 103).

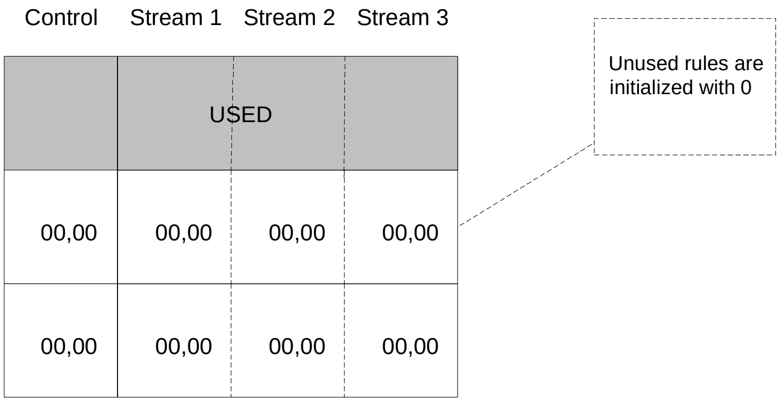


Figure 103: Unused Rows and Rules Must Be Initialized with Zeros

Frame patterns are stored as data structures in memory. A control word is always present in a 64 bit entry/row. The control word describes proceeding data fields in the entry.

In 10/100 Mbps mode, one WOL entry requires three 64-bit wide rows (see Table 117). The total length of an entry is 192 bits. Each 64-bit row contains a 16-bit control word, which identifies byte enables (see Table 118). The remaining 48-bits contains 2-byte rules. The 2-byte rules are distributed across three streams: S, S+1, and S+2. The next row's 2-byte rules will correspond to three more streams: S+3, S+4, and S+5. Both Table 117 and Table 118 use Sx notation to denote separate comparison streams. The D0 notation indicates the first 2 bytes in the packet stream are compared.

Table 117: 10/100 Mbps Mode Frame Patterns Memory

63 48	*47 32*	*31 16*	*15 0*
CTRL012	S0D0	S1D0	S2D0
CTRL345	S3D0	S4D0	S5D0
CTRL678	S6D0	S7D0	S8D0

Table 118: Frame Control Field for 10/100 Mbps Mode

Bits	Field	Description	Access
63-62	Reserved		
61	S0 High Byte Enable	Enable S0 higher byte for comparison	R/W
60	S0 Low Byte Enable	Enable S0 lower byte for comparison	R/W
59	S1 High Byte Enable	Enable S1 higher byte for comparison	R/W
58	S1 Low Byte Enable	Enable S1 lower byte for comparison	R/W
57	S2 High Byte Enable	Enable S2 higher byte for comparison	R/W
56	S2 Low Byte Enable	Enable S2 lower byte for comparison	R/W
55-51	Reserved		
50	S0 Done	End of S0 Stream	R/W
49	S1 Done	End of S1 Stream	R/W
48	S2 Done	End of S2 Stream	R/W

Table 119 shows an example of how 10/100 Mbps frame data is split up in the pattern data structure. Nine streams are compared simultaneously with three 64-bit rows comprising one WOL entry. Rows 0-2 compare frame data0 against the first 2 bytes of nine streams. Rows 3-5 compare frame data1 against the next 2 bytes of nine pattern streams. Rows 6-9 compare data2 against the last 2 bytes of nine pattern streams. The nine rules of each WOL entry (three 64-bit rows) may be uniquely defined.

Table 119: Example of Splitting 10/100 Mbps Frame Data in Pattern Data Structure

Data[63:48]	Data[47:32]	Data[31:16]	Data[15:0]
Control Bits	Stream 0 data 0	Stream 1 data 0	Stream 2 data 0
Control Bits	Stream 3 data 0	Stream 4 data 0	Stream 5 data 0
Control Bits	Stream 6 data 0	Stream 7 data 0	Stream 8 data 0
Control Bits	Stream 0 data 1	Stream 1 data 1	Stream 2 data 1
Control Bits	Stream 3 data 1	Stream 4 data 1	Stream 5 data 1
Control Bits	Stream 6 data 1	Stream 7 data 1	Stream 8 data 1
Control Bits	Stream 0 data 2	Stream 1 data 2	Stream 2 data 2
Control Bits	Stream 3 data 2	Stream 4 data 2	Stream 5 data 2
Control Bits	Stream 6 data 2	Stream 7 data 2	Stream 8 data 2

Firmware Mailbox

When the BCM57XX family initializes (the firmware bootcode is loaded from NVRAM when the chip powers on or when reset completes), the BCM57XX firmware bootcode checks the T3_FIRMWARE_MAILBOX in shared memory. When the T3_MAGIC_NUM signature (0x4b455654) is present, the bootcode does not issue a hard reset to the PHY. This is especially important in WOL mode since the PHY should not be reset.

Before the host software issues a reset to the BCM57XX, it must write the T3_MAGIC_NUM to the shared memory address T3_FIRMWARE_MAILBOX (0xb50). This address is a software mailbox, which bootcode polls before it resets the PHY. The bootcode will acknowledge the signature by writing the one's complement of the T3_MAGIC_NUM back into the T3_FIRMWARE_MAILBOX. If the T3_MAGIC_NUM is present, the bootcode will not reset the PHY. After resetting the BCM57XX, host software should poll for the one's complement of the T3_MAGIC_NUM before it proceeds, otherwise, bootcode initialization may interfere with the host software initialization.

If the host software will be controlling the WOL configuration, it should write the DRV_WOL_SIGNATURE (0x474c0000) to the shared memory address DRV_WOL_MAILBOX (0xd30) so that the bootcode will not take over the WOL initialization. If the DRV_WOL_SIGNATURE is not present, and WOL has been enabled, the bootcode will assume that the host software is a legacy driver and skip the WOL initialization. If WOL is disabled, the bootcode will take over the WOL initialization based on the NVRAM configuration.

Table 120: Firmware Mailbox Initialization

Name	Address	Recommended Value
T3_FIRMWARE_MAILBOX	0x0B50	0x4B455654
DRV_WOL_MAILBOX	0xd30	0x474c0000

BCM5401 Auto-negotiation

The BCM5401 PHY and BCM57XX integrated PHY cores should be configured to auto-negotiate for a 10 Mbps connection (see Table 121). This step is required if the NIC needs to be placed into a D3 cold state. Half- or full-duplex operation is acceptable. Software must modify auto-advertise configurations in the BCM5401's MDI registers. The link partner will read advertisement settings to find a highest common capability. Since WOL requires 10 Mbps wire speed, the two PHYs will effectively auto-negotiate for half- or full-duplex connection (see "1000BASE-X Auto-Negotiation" on page 245 for a detailed discussion on auto-negotiation).

Table 121: Recommended Settings for PHY Auto-Negotiation

Register	Bit	Recommended Value
Auto_Negotation_Advertisement	10_BASE_TX_Half_Duplex	Enable
Auto_Negotation_Advertisement	10_BASE_TX_Full_Duplex	Enable
Auto_Negotation_Advertisement	100_BASE_TX_Half_Duplex	Disable
Auto_Negotation_Advertisement	100_BASE_TX_Full_Duplex	Disable
1000BASE-T_Control	1000_BASE_TX_Half_Duplex	Disable
1000BASE-T_Control	1000_BASE_TX_Full_Duplex	Disable

BCM5401 Power Management (BCM5700 Only)

Configurations for WOL mode are listed in Table 122.

Table 122: WOL Mode Configuration

Register	Bit	Recommended Value
Auxilary_Control	Wake_On_LAN	Enable

BCM57XX **Power Management**

The clocking inputs need to be modified for WOL mode (see Table 123). The RX and TX CPU are not required during WOL operation, so their clocks can be disabled. The MAC has an internal phase-locked loop that clocks internal logic at 133 MHz. Software must select an alternate clocking source and then disable this PLL.

Table 123: WOL Mode Clock Inputs

Register	Bit	Recommended Value
PCI Clock_Control	TX RISC_Clock_Disable	Enable
PCI Clock_Control	RX RISC_Clock_Disable	Enable
PCI Clock_Control	Select_Alternate_Clock	Enable
PCI Clock_Control	PLL133	Enable

The settings shown in Table 124 enable Magic Packet detection logic in the MAC. These setting also enable the MAC to assert PME on the PCI bus. The RX MAC should maintain the multicast and broadcast settings that were previously configured by the NOS. The Microsoft power management specification states:

"Only a frame that passes the device's MAC, broadcast, or multicast address filter and matches on the previously loaded sample patterns will cause the wake-up signal to be asserted."

The ACPI_Power-on bit needs to be set for pattern match, but not for Magic Packet recognition. The Magic Packet detection mechanism is separate from the pattern match mechanism. Host Software may configure WOL using four filter permutations:

- Pattern match WOL disabled. Magic Packet disabled.
- Pattern match WOL enabled. Magic Packet disabled.
- Pattern match WOL disabled. Magic Packet enabled.
- Pattern match WOL enabled. Magic Packet enabled.

Table 124: Magic Packet Detection Logic Enable

Register	Bit(s)	Recommended Value
PCI Power_Management_Control/Status	PME_Enable	Enable
PCI Power_Management_Control/Status	Power_State	0x03
Ethernet_MAC_Mode	ACPI_Power-On	See above
Ethernet_MAC_Mode	Magic_Packet_Detection	See above

Register Quick Cross Reference

BCM5401 and BCM57XX Integrated PHYs

Table 125 lists the WOL mode control registers in the BCM5401 and BCM57XX Integrated PHYs.

Table 125: PHY WOL Mode Control Registers

MDI Register	Bit(s) Name	Description	Cross Reference
Auto_Negotation_Advertisement	10_BASE_TX_Half_Duplex	Advertise to link partner that local PHY is capable of 10 Mbps half-duplex operation.	See BCM5401 Data Sheet or "Auto-Negotiation Advertisement Register (PHY_Addr = 0x1, Reg_Addr = 04h)" on page 608.
	10_BASE_TX_Full_Duplex	Advertise to link partner that local PHY is capable of 10 Mbps full-duplex operation.	
	100_BASE_TX_Half_Duplex	Advertise to link partner that local PHY is capable of 100 Mbps half-duplex operation.	
	100_BASE_TX_Full_Duplex	Advertise to link partner that local PHY is capable of 100 Mbps full-duplex operation.	
1000BASE-T_Control	1000_BASE_TX_Half_Duplex	Advertise to link partner that local PHY is capable of 1000 Mbps half-duplex operation.	See BCM5401 Data Sheet or "1000BASE-T Control Register (PHY_Addr = 0x1, Reg_Addr = 09h)" on page 615.
	1000_BASE_TX_Full_Duplex	Advertise to link partner that local PHY is capable of 1000 Mbps full-duplex operation.	
Auxilary_Control	Wake_On_LAN	Enable Wake On LAN capability with low-power consumption.	See BCM5401 Data Sheet or "Auxiliary Control Register (PHY_Addr = 0x1, Reg_Addr = 18h, Shadow = 000, Normal)" on page 645.

Integrated MACs

Table 126 lists the WOL mode control registers in the BCM57XX integrated MACs.

Table 126: Integrated MAC WOL Mode Control Registers

Register	*Bit(s) Name*	*Description*	*Cross Reference*
WOL_Pattern_ Pointer	All	This register points to an internal memory location. Programmers should calculate pointer value by dividing a base address by 8.	See "WOL Pattern Pointer Register (Offset 0x430)" on page 385.
WOL_Pattern_ Configuration	Length	The number of memory arbiter clock cycles needed to read X bytes in the RX stream/frame.	See "WOL Pattern Configuration Register (Offset 0x434)" on page 386.
	Offset	The number of bytes into the RX stream/frame to begin the pattern comparison.	
Ethernet_MAC_ Mode	Port_Mode	This bit field specifies the type of interface the BCM57XX port is currently using: MII, GMII, TBI, or none.	See "Ethernet MAC Mode Register (Offset 0x400)" on page 377.
	Link_Polarity	Flip polarity of link inputs.	
	Magic_Packet_ Detection	Enable WOL pattern filtering.	
	Promiscuous_mode	All frames are forwarded, without any filtering, when this bit is enabled.	
PCI Clock_Control	TX RISC_Clock_Disable	Disable the clock to the transmit CPU.	See "PCI Clock Control Register (Offset 0x74)" on page 334.
	RX RISC_Clock_Disable	Disable the clock to the receive CPU.	
	Alternate_Clock_Source	Use an alternate clock as a reference, rather than the PLL 133.	
	PLL133	Disable the 133-MHz phase-locked loop.	
Misc Local Control	Misc_Pin_0_Output	GPIO pin 0.	See "Miscellaneous Local Control Register (Offset 0x6808)" on page 507.
	Misc_Pin_0_Output_ Enable	When asserted, MAC drives pin output.	
	Misc_Pin_1_Output	GPIO pin 1.	
	Misc_Pin_1_Output_ Enable	When asserted, MAC drives pin output.	
	Misc_Pin_2_Output	GPIO pin 2.	
	Misc_Pin_2_Output_ Enable	When asserted, MAC drives pin output.	
Power Management Control/Status	PME_Enable	Enable the BCM57XX family to assert PME on PCI bus.	See "Power Management Control/Status Register (Offset 0x4C)" on page 318.
	Power_State	Set the ACPI power state: D0, D3.	

WOL Data Flow Diagram

The BCM57XX family and PHY are both configured for WOL mode. The process is as follows:

1. Clear the PME_Status bit in the "PMCSR-BSE Register (Offset 0x4E)" on page 318. This bit must be cleared, so the PME interrupt is not immediately generated once the NIC is moved to the D3 state. The bit could be asserted from a previous D3-D0 transition.
2. Set the Mask_PCI_Interrupt_Output bit in the Miscellaneous_Host_Control register (see "Miscellaneous Host Control Register (Offset 0x68)" on page 325). This bit should be set, so the BCM57XX family does not generate interrupts during the WOL configuration of the PHY. The device driver's ISR may attempt to reset and reconfigure the PHY as part of an error recovery code path.
3. If Host software needs to place the NIC into D3 cold state, the following step is necessary. Set the 10_Base_TX_Half_Duplex and 10_BASE_TX_Full_Duplex Capability bits, in the "Auto-Negotiation Advertisement Register (PHY_Addr = 0x1, Reg_Addr = 04h)" on page 608. Clear the 100_BASE_TX_Full_Half_Duplex and 100_BASE_TX_Full_Duplex Capability bits, in the "Auto-Negotiation Advertisement Register (PHY_Addr = 0x1, Reg_Addr = 04h)" on page 608. Clear the 1000_BASE_TX_Half_Duplex and 1000_BASE_TX_Full_Duplex Capability bits, in the "1000BASE-T Control Register (PHY_Addr = 0x1, Reg_Addr = 09h)" on page 615. The BCM5401's link partner will now only be able to auto-negotiate for 10 Mbps speed full/half-duplex.
4. Set the Restart_Auto_Negotiation bit in the "MII Control Register (PHY_Addr = 0x1, Reg_Addr = 00h)" on page 604. The integrated PHY and link partner will now reconfigure for 10 Mbps wire speed. Essentially, 10 Mbps link must be auto-negotiated or forced.
5. Disable the FHDE, RDE, TDE bits of the "Ethernet MAC Mode Register (Offset 0x400)" on page 377", and on-chip RISCs.
6. Host software must write the signature 0x4B455654 to internal memory address 0x0B50. Check for one's complement of 0x4B455654
7. Enable the Wake_On_LAN bit in the "Auxiliary Control Register (PHY_Addr = 0x1, Reg_Addr = 18h, Shadow = 010, Power Control)" on page 651.
8. For Interesting Packet WOL Only: Set up the Interesting Packet pattern in BCM57XX device local memory.
9. For Interesting Packet WOL Only: Write a pointer value to the "WOL Pattern Pointer Register (Offset 0x430)" on page 385. This register uses a normalized pointer value, not a device base address. The value written to this register is BCM5700_BASE_ADDR/8. The base address must be a specific location in local memory: 0x8000, 0xC000, or 0xD000. The choice of memory location depends upon other MAC configurations, and the selection is not arbitrary.
10. For Interesting Packet WOL Only: Write the Offset field in the "WOL Pattern Configuration Register (Offset 0x434)" on page 386. The WOL pattern checker will position into received frames on two-byte intervals. The pattern checker compares two bytes in parallel, so host software should program the offset field accordingly. Host software may perceive this unit as OFFSET_BYTE/2 units.
11. For Interesting Packet WOL Only: Write the Length field in the "WOL Pattern Configuration Register (Offset 0x434)" on page 386. The length value is specified in terms of Memory Arbiter clock cycles, not bytes/words/dwords. A comprehensive discussion of how the clock cycles are calculated will be presented.
12. Set the Port_Mode field in the "Ethernet MAC Mode Register (Offset 0x400)" on page 377 to GMII mode. These bits enable the GMII between the BCM57XX family and the BCM5401 physical layer. Enable the Link_Polarity bit in the "Ethernet MAC Mode Register (Offset 0x400)" on page 377. This bit will set the polarity of the LNKRDY signal between the physical layer and MAC. On the BCM57XX reference NIC, LNKRDY is tied to the 10 Mbps active low signal routed from the BCM5401 PHY. Host software should set this bit, so the MAC may detect 10Mpbs link.
13. For Interesting Packet WOL Only: Enable the ACPI_Power-On bit in the "Ethernet MAC Mode Register (Offset 0x400)" on page 377. This bit will enable logic for D3 hot/cold transitions to D0 ACPI state. The MAC will also be capable of asserting PME on the PCI bus.
14. For Magic Packet WOL Only: Enable the Magic_Packet_Detection bit in the "Ethernet MAC Mode Register (Offset 0x400)" on page 377. The WOL logic will compare RX frames for Magic Packet patterns.

15. Set the RX RISC_Clock_Disable bit in the PCI Clock_Control register (see "PCI Clock Control Register (Offset 0x74)" on page 334). The receive CPU will be stopped, and the clocking circuitry disabled. Set the TX RISC_Clock_Disable bit in the "PCI Clock Control Register (Offset 0x74)" on page 334. The transmit CPU will be stopped and the clocking circuitry disabled.
16. Set the Enable_Alternate_Clock bit in the PCI Clock_Control register (see "PCI Clock Control Register (Offset 0x74)" on page 334). The BCM57XX family's 133 MHz Phase Locked Loop (PLL) no longer clocks internal logic and an alternate clock reference is used. Set the PLL LowPowerClock bit while keeping the Enable_Alternate_Clock bit set. Wait at least 27 µs and then clear the Enable_Alternate_Clock bit. The BCM57XX family's PLL will be switched to its lower power consumption mode.
17. In NIC applications, switch from VMAIN to VAUX in order to prevent a GRC reset. Set the required GPIOs of BCM57XX if any of them are used for switching the power from VMAIN to VAUX.
18. Enable the RX MAC by setting the Enable bit of "Receive MAC Mode Register (Offset 0x468)" on page 391" and put it in promiscuous mode by setting the Promiscuous Mode bit of "Receive MAC Mode Register (Offset 0x468)" on page 391".
19. Enable the PME bit in the PCI "PMCSR-BSE Register (Offset 0x4E)" on page 318. Set the Power_State bits to D3 in the "PMCSR-BSE Register (Offset 0x4E)" on page 318.

FLOW CONTROL

DESCRIPTION

The BCM57XX family supports IEEE 802.3x flow control. Flow control is a switched Ethernet capability, where link partners may pause traffic. 802.3x flow control specifies that a MAC sublayer may transmit pause frames. The pause frames instruct the MAC's link partner to wait a specified amount of time, before sending additional frames. This delay provides the MAC time to free packet buffers. Conversely, the MAC sublayer must also accept/receive pause frames. Flow control is used by switches and bridges to prevent clients of dissimilar speeds from exhausting switching packet buffers. Clients and servers may use flow control for similar reasons. A very important requirement is that both link partners must share a full-duplex connection for flow control to be enabled. 802.3x flow control does not operate on a half-duplex connection. More information on flow control can be found in Appendix A "Flow Control" on page 703.

The following architectural blocks are integral to flow control:

- Transmit MAC
- Receive MAC
- Statistics Block
- PHY Auto-negotiation
- PHY Auto-Advertise

Operational Characteristics

The BCM57XX family implements pause functionality using Xon and Xoff states. The MAC will extract a pause quantum from a pause control frame. Then, the MAC will configure its internal timer with the pause_time specified by the link partner. Frames that are currently in the transmit engine will be completed before the transmit engine is inhibited. The MAC has moved flow control into a Xoff state once the transmit engine is inhibited. Note that the transmit engine is not completely disabled since the IEEE 802.3 specification stipulates that MAC control frames should not be paused.

One of the following conditions will move the BCM57XX family into a Xon state:

- Link partner sends a pause frame with pause_time = 0.
- Internal pause timer expires.

Transmit MAC

The transmit MAC is responsible for sending flow control frames. Software enables the transmit MAC to send flow control frames by setting the Enable_Flow_Control bit in the Transmit_MAC_Mode register (see "Transmit MAC Mode Register (Offset 0x45C)" on page 390). When software clears the Enable_Flow_Control bit, the transmit MAC will not generate flow control frames. The MAC_RX_MBUF_Low_Water_Mark register value (see "MAC RX MBUF Low Watermark Register (Offset 0x4414)" on page 470) triggers PAUSE frames to be transmitted when a threshold value is passed. Software may alter the watermark to tune system performance. The watermark recommendation in Table 127 assumes no external SSRAM.

Table 127: Transmit MAC Watermark Recommendation

Register	*Recommended Value*
MAC_RX_MBUF_Low_Water_Mark	24

As soon as PAUSE frame is transmitted, any incoming packet can be dropped, and the ifInDiscard counter in statistics (see "Interface Statistics" on page 108) will increase. When packet size is small (64 bytes) with 1000 Mbps, more frames can be dropped. Even if the PAUSE frame is transmitted, Pause frames cannot inhibit MAC control frames.

Low Water Mark Maximum Receive Frames register (see "Low Watermark Maximum Receive Frames Register (Offset 0x504)" on page 394) control the number of good frames to receive after the RX MBUF Low Water Mark has been reached. After the RX MAC receives this number of frames, it will drop subsequent incoming frames until the MBUF High Water Mark is reached.

The 802.3 pause control frame contains a pause_time field. The BCM57XX transmit MAC will insert a time quanta into the pause_time field. Software should set the Enable_Long_Pause bit in the Transmit_MAC_Mode register to configure long pause quanta. Clearing the Enable_Long_Pause bit will default the pause_time back to the shorter quanta. Table 128 shows the pause quanta based on the Enable_Long_Pause bit setting.

Table 128: Pause Quanta

Enable_Long_Pause Bit	*Pause_Time*
DISABLED (0)	0x1FFF
ENABLED (1)	0xFFFF

Receive MAC

The BCM57XX receive MAC's link partner may wish to inhibit frame transmission until upstream resources become available. The receive MAC must be configured to accept 802.3x pause frames (see Table 129). Software should set the Enable_Flow_Control bit in the Receive_MAC_Mode_Control register to enable automatic processing of flow control frames. If software clears the Enable_Flow_Control bit, 802.3x pause frames will be discarded. The Keep_Pause bit in the Receive_MAC_Mode_Control register will instruct the RX engine to forward pause frames to host memory. Software may be interested in setting this bit for debugging or promiscuous/sniffer configurations. Passing pause frames to the host will increase DMA and protocol processing and consume available host buffers. The receive MAC will filter pause control frames when the Keep_Pause bit is disabled.

Table 129: Keep_Pause Recommended Value

Register.Bit	*Recommended Value*
Receive_MAC_Mode_Control.Keep_Pause	DISABLED

Statistics Block

The statistic block shown in Table 130 is a common data structure. The relationships of flow control statistics are discussed in this section. Xon/Xoff statistical counters are related to internal BCM57XX flow control states. Xon is associated to transmit enabled state and Xoff is associated to transmit disabled state. These Xon/Xoff states are not part of the IEEE 802.3 specification; the BCM57XX family uses Xon/Xoff to manage flow control state and transitions. The Xon/Xoff statistics provide programmers with a high level of granularity for the measurement of BCM57XX flow control performance in a LAN (see Appendix A "Flow Control" on page 703).

Table 130: Statistic Block

Statistic	*Description*
xoffStateEntered	This counter is bumped under the following conditions: 1 802.3 MAC flow control pause frame received with valid CRC. 2 (Pause_time > 0) The link partner requests transmission inhibit. The counter increments independently of the enabled/disabled state of Receive_MAC_Mode_Control.Flow_Enabled.
xonPauseFramesReceived	This counter is incremented under the following conditions: 1 802.3 MAC flow control pause frame received with valid CRC. 2 (Pause_time == 0) The link partner no longer requires the BCM57XX family to pause/wait/delay outgoing packets. The counter increments independently of the enabled/disabled state of Receive_MAC_Mode_Control.Flow_Enabled.
xoffPauseFramesReceived	This counter is incremented under the following conditions: 1 802.3 MAC flow control pause frame received with valid CRC. 2 (Pause_time > 0) The link partner requires the BCM57XX family to pause/wait/delay outgoing packets. The counter increments independently of the enabled/disabled state of Receive_MAC_Mode_Control.Flow_Enabled.
outXon	This counter is incremented under the following conditions: 1 Transmit_MAC_Mode_Control.Flow_Enabled bit is set. 2 (MAC_RX_MBUF_Low_Water_Mark > Threshold Value MAC resources are available. 3 (pause_time == 0) 802.3 MAC flow control frame is sent.

Table 130: Statistic Block (Cont.)

Statistic	*Description*
outXoff	This counter is incremented under the following conditions: 1 Transmit_MAC_Mode_Control.Flow_Enabled bit is set. 2 (MAC_RX_MBUF_Low_Water_Mark < Threshold Value) MAC resources are running low and a pause is desired. 3 (pause_time > 0) 802.3 MAC flow control frame is sent.

PHY Auto-Negotiation

A full description of software's role in MAC auto-negotiation can be found in "1000BASE-X Auto-Negotiation" on page 245. The PHY encodes flow control capability into Fast Link Pulse (FLPs) bursts. Link partners will extract encoded flow control capability from FLPs and then create a Link Code Word (LCW). The LCW is a message, which contains a selector and technology ability field. The technology ability field contains a bit called Pause_Operation_for Full_Duplex_Link (A5). Refer to Annex 28-B of the 802.3 specifications. The A5 bit signifies that a link partner has implemented pause functionality. If both link partners support auto-negotiation, they will further exchange data regarding flow control, using the next page bit in the LCW.

Auto-advertise is integrally tied to auto-negotiation. If link partner does not support pause functionality, the PHY Auto_Negotation_Link_Partner_Ability_Register will not have the Pause_Capable bit set. The BCM57XX family should not send pause frames to this link partner since flow control is not implemented or disabled. The BCM57XX family can still accept pause frames, but sending a pause frame will not yield a desired result.

Register Quick Cross Reference

BCM5401 and BCM57XX Integrated PHYs

Table 131 lists the flow control registers in the BCM5401 and BCM57XX integrated PHYs.

Table 131: PHY Flow Control Registers

MDI Register	*Bit(s) Name*	*Description*	*Cross Reference*
MII_status	Link_Status	Link Pass State which indicates if a valid link has been established.	BCM5401 Data Sheet or "MII Status Register (PHY_Addr = 0x1, Reg_Addr = 01h)" on page 606
MII_auxiliary_status	Auto_Negotation_HCD	Current Operating Mode and Speed	BCM5401 Data Sheet or "Auxiliary Status Summary Register (PHY_Addr = 0x1, Reg_Addr = 19h)" on page 657
Auto-negotiation Advertisement	Asymetric_Pause	Advertise to link partner, that asymmetric pause is desired. This bit works in conjunction with Pause_Capable bit.	See BCM5401 Data Sheet or "Auto-Negotiation Advertisement Register (PHY_Addr = 0x1, Reg_Addr = 04h)" on page 608.
	Pause_Capable	The pause capable bit indicates whether half/full-duplex pause is advertised.	

Table 131: PHY Flow Control Registers (Cont.)

MDI Register	Bit(s) Name	Description	Cross Reference
Auto-negotiation Link Partner Ability	Asymetric_Pause Pause_Capable	Link Partner capability—the partner desires asymmetric pause. Link partner capability—the partner is capable of full or half-duplex pause.	See BCM5401 Data Sheet or "Auto-Negotiation Link Partner Ability Register (PHY_Addr = 0x1, Reg_Addr = 05h)" on page 610.

Integrated MACs

Table 132 lists the flow control registers in the BCM57XX integrated MACs.

Table 132: Integrated MAC Flow Control Registers

Register	Bit(s) Name	Description	Cross Reference
Receive MAC Mode	Enable_Flow_Control	Enable automatic processing of 802.3 flow control frames.	See "Receive MAC Mode Register (Offset 0x468)" on page 391.
Transmit MAC Mode	Enable_Flow_Control	Enable automatic processing of 802.3 flow control frames.	See "Transmit MAC Mode Register (Offset 0x45C)" on page 390.
MAC_RX_MBUF_Low_Water_Mark	All 32 bits	The number of internal buffers that must be available before the RX engine can accept a frame from the wire. Threshold value for initiating flow control	See "MAC RX MBUF Low Watermark Register (Offset 0x4414)" on page 470.

Flow Control Initialization Pseudocode

```
//Check the Link State
If (MII_Status_Reg.Link_Status == TRUE) Then
{
   //Check PHY status register for full-duplex configuration
   If (MII_Aux_Status_Reg.Auto_Neg_HCD ==
      (1000_FULL_DUPLEX Or 100_FULL_DUPLEX Or 10_FULL_DUPLEX) ) Then
   {
         //Check if USER has forced either auto-negotiation or auto-advertise
         If ( (Driver_Auto_Neg_Variable == ENABLED) And
            (Driver_Auto_Advertise_Variable != FORCED_SPEED_DUPLEX )) Then
         {
            // Probe Phy control registers for advertised flow control info
            // Expected abilities should match the configured abilities. Expected abilities
            // are based on the IEEE 803.3ab flow control subsection.
            If  ( (Auto_Neg_Advertise_Reg.Asymetric_Pause != 802.3ab_Table_28B-3 ) And
               (Auto_Neg_Advertise_Reg.Pause_Capable != 802.3ab_Table_28B-3 ) ) Then
            {
               //The current advertised state does not match 802.3 specifications
               Driver_ Link__link_state = LINK_STATUS_DOWN
            }
            Else
            {
               If  (Auto_Neg_Advertise_Reg.Pause_Capable == ENABLED)
               {
                  If ( Auto_Neg_Advertise_Reg.Asymetric_Pause == ENABLED) ) Then
                  {
                     If  (Auto_Neg_Link_Partner_Ability_Reg.Pause_Capable == ENABLED)
Then
                     {
                        Driver_Flow_Capability = FLOW_CONTROL_TRANSMIT_PAUSE \
                           |  FLOW_CONTROL_RECEIVE_PAUSE
                     }
                     Else If (Auto_Neg_Link_Partner_Ability_Reg.Asymetric_Pause == \
                        ENABLED) Then
                     {
                        Driver_Flow_Capability = FLOW_CONTROL_RECEIVE_PAUSE
                     }
                     Else
                     {
                        Driver_Flow_Capability = NONE
                     }
                  }
                  //The local physical layer was not configured to advertise Asymmetric
pause
                  Else
                  {
                     If  (Auto_Neg_Link_Partner_Ability_Reg.Pause_Capable == ENABLED)
Then
                     {
                        Driver_Flow_Capability = FLOW_CONTROL_TRANSMIT_PAUSE \
                           | FLOW_CONTROL_RECEIVE_PAUSE
                     }
                     Else
                     {
                        Driver_Flow_Capability = NONE
                     }
                  }
               }
               // The local physical layer was not configured to advertise Pause capability
               Else If  (Auto_Neg_Advertise_Reg.Asymetric_Pause == ENABLED) Then
               {
                  If  (Auto_Neg_Link_Partner_Ability_Reg.Pause_Capable == ENABLED) Then
                  {
                     Driver_Flow_Capability = FLOW_CONTROL_TRANSMIT_PAUSE
                  }
                  Else
                  {
                     Driver_Flow_Capability = NONE
                  }
               }
         } //Link Status is up
      } // Auto negotiation was not disabled && Speed Duplex was not forced
      Else
      {
         // The use forced speed/duplex, so the partner's flow control capabilities are
         // indeterminate - software cannot use the Link_Partner_Abitity
         // registers.
         Driver_Flow_Capability= DISABLED
      }
   } //The current link is full-duplex at 10/100/1000 wire speeds
   Else
   {
         //Full-Duplex mode is not available or forced half-duplex
         //Flow control is not available in half-duplex mode.
         Driver_Flow_Capability = NONE
   }
   //Configure MAC Flow Control Registers
   if ( Driver_Flow_Capability & FLOW_CONTROL_RECEIVE_PAUSE )
   {
      Receive_MAC_Mode_Control_Register.Enable_Flow_Control = ENABLED
   }
   if ( Driver_Flow_Capability & FLOW_CONTROL_TRANSMIT_PAUSE ) Then
   {
      Transmit_MAC_Mode_Control_Register.Enable_Flow_Control = ENABLED
   }
}  // Link is up on the local PHY
```

Section 11: Interrupt Processing

HOST COALESCING

It is well known that interrupt coalescing (or interrupt moderation) is a common technique that can be used by NIC vendors to increase the performance of NIC. Some high-level descriptions of the benefits of interrupt coalescing can be found at:

- http://www.microsoft.com/HWDEV/devdes/optinic.htm
- http://support.microsoft.com/support/kb/articles/Q170/6/43.ASP
- http://msdn.microsoft.com/library/books/serverdg/networkadapterrequirements.htm

DESCRIPTION

The BCM57XX family supports the concept of host coalescing. Host coalescing controls when status information is returned to the host, and when interrupts are generated. The BCM57XX family provides a number of SW configurable registers that control when/how the BCM57XX family updates the host with status information and how often the BCM57XX family asserts an interrupt.

When the BCM57XX family has completed transmit or receive events, it will update a Status block in host memory. This status block contains information that tells the host which transmit buffers have been DMAed by the NIC, and which receive Buffer Descriptors (BDs) have been consumed by a newly arrived received packet. Normally, the host will check this status block whenever an interrupt is generated. In addition, the host could also poll the status block to determine whether or not it had been updated by the hardware since the last time the host had read the status block (this is called during interrupt processing).

Whenever the NIC updates the status block, it will make a decision about whether to assert the interrupt line ($\overline{\text{INTA}}$) or not. The BCM57XX family has special interrupt avoidance mechanisms that allow the host to tell the NIC not to generate an interrupt when it writes a status block back to the host. In addition, there are also mechanisms that allow host SW to control when and how often the status block is updated. For instance, the host could configure the NIC to only update status block after it receives two packets, as opposed to one packet. These mechanisms are documented in more detail below.

OPERATIONAL CHARACTERISTICS

The BCM57XX family DMAs the status block (see "Status Block Base Address Register (Offset 0x3C44)" on page 457) to host memory before a line interrupt or MSI is generated. The host ISR reads the update bit at the top of the status block and checks whether this bit is set to 1 or not. When set to 1, the updated bit of status block indicates the host that the status block has been refreshed by the MAC. The ISR must then write to clear/de-assert this bit to dirty the status block, and then the ISR may proceed to read the updated producer/consumer index pointers. This mechanism allows host system software to determine if the status block has been updated. Due to various asynchronous timing issues (dependent upon platform) the ISR may occasionally see stale data. The ISR may either spin and wait for the status block DMA to complete and explicitly flush the status block or just wait for the next line interrupt.

REGISTERS

The BCM57XX family supports a variety of registers that affect status block updates and interrupt generation (see Table 133). Each BCM57XX Data Sheet defines the register layout and bit fields; however, software requires a robust description of the register functionality.

Table 133: Interrupt-related Registers

Register	Cross Reference
Miscellaneous Host Control register. The two bits of this register that are related to interrupts are: • Mask PCI Interrupt Output (aka Mask Interrupt) bit • Clear Interrupt $\overline{\text{INTA}}$ bit	See "Miscellaneous Host Control Register (Offset 0x68)" on page 325.
Miscellaneous Local Control register. The two bits of this register that are related to interrupts are: • Set Interrupt bit • Clear Interrupt bit	See "Miscellaneous Local Control Register (Offset 0x6808)" on page 507.
Interrupt Mailbox 0 register	See "Interrupt Mailbox 0 Register (Offset 0x200)" on page 372 for host standard and flat modes and "Interrupt Mailbox 0 Register (Offset 0x5800)" on page 492 for indirect mode.
Interrupt Mailboxes 1-7 register	See "Other Interrupt Mailbox Registers (Offset 0x208-0x218)" on page 373 for host standard and flat modes and "Other Interrupt Mailbox Registers (Offset 0x5808-0x5818)" on page 492 for indirect mode.
Receive Coalescing Ticks register	See "Receive Coalescing Ticks Registers (Offset 0x3C08)" on page 453.
Send Coalescing Ticks register	See "Send Coalescing Ticks Register (Offset 0x3C0C)" on page 453.
Receive Coalescing Ticks During Interrupt register	See "Receive Coalescing Ticks During Interrupt Register (Offset 0x3C18)" on page 455.
Send Coalescing Ticks During Interrupt register	See "Send Coalescing Ticks During Interrupt Register (Offset 0x3C1C)" on page 455.
Receive Max Coalesced BD Count register	See "Receive Max Coalesced BD Count (Offset 0x3C10)" on page 454.
Send Max Coalesced BD Count register	See "Send Max Coalesced BD Count (Offset 0x3C14)" on page 454.
Receive Max Coalesced BD Count During Interrupt register	See "Receive Max Coalesced BD Count During Interrupt (Offset 0x3C20)" on page 456.
Send Max Coalesced BD Count During Interrupt register	See "Send Max Coalesced BD Count During Interrupt (Offset 0x3C24)" on page 456.

MSI

PCI Specification 2.2 defines a new mechanism for a device to request services by its device driver. It is called Message Signaled Interrupt (MSI). MSI will eventually deprecate traditional interrupt mechanism. In MSI, device DMAs a specified DWORD data to a specified host address if it needs to request services by its device driver. The MSI state machine can be enabled/disabled by setting/resetting the Enable bit of MSI Mode register (offset 0x6000). By default, this bit is set to 1 indicating that the MSI state machine is enabled. The main advantages of MSI generation versus using a traditional interrupt are as follows:

- Eliminates the need for interrupt signal trace on the PCI device.
- Eliminates the need to perform a dummy read from the device by the device driver in its interrupt service routine. This is done to force all posted memory writes to be flushed to the host memory.

Traditional Interrupt Scheme

A simplified block diagram showing traditional interrupt scheme is depicted in Figure 104.

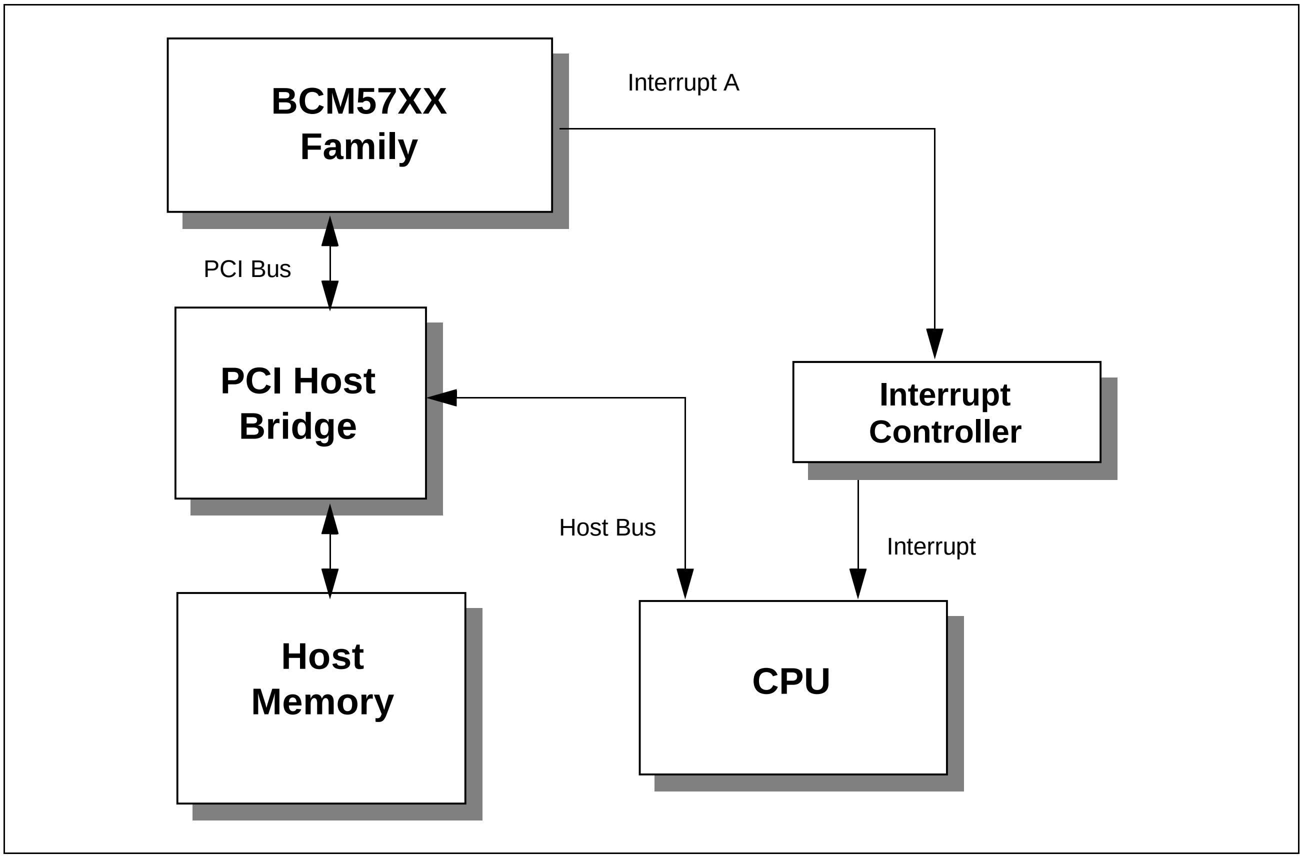


Figure 104: Traditional Interrupt Scheme

To clarify second issue in traditional interrupt scheme, an example is given. The BCM57XX family receives one or more packets from the networks. The BCM57XX family does the following:

- DMAs data of received packets to the host.
- DMAs receive buffer descriptors to Receive Return Ring in the host memory.
- DMAs status block to the host memory.
- Generates an interrupt to request its device driver for processing.

The writes are posted and are actually performed at some later time by the PCI Host Bridge. When interrupt service routine of device driver is executed, the driver reads the status block from the host memory and finds that status block does not contain latest index information if the writes for status block are not performed by the PCI Host Bridge yet. The scheme to resolve this problem is to do a dummy read of the BCM57XX family in the beginning of the interrupt service routine. The dummy read has to traverse the same bridge that memory writes from the BCM57XX family have to traverse to get to the host memory. The ordering rules for bridges dictate that the bridge must flush its posted write buffers before permitting a read to traverse the bridge. As a result, writes for status block are flushed to the host memory by the bridge before dummy read cycle is completed.

MESSAGE SIGNALED INTERRUPT

A simplified block diagram showing a possible MSI scheme is depicted in Figure 105.

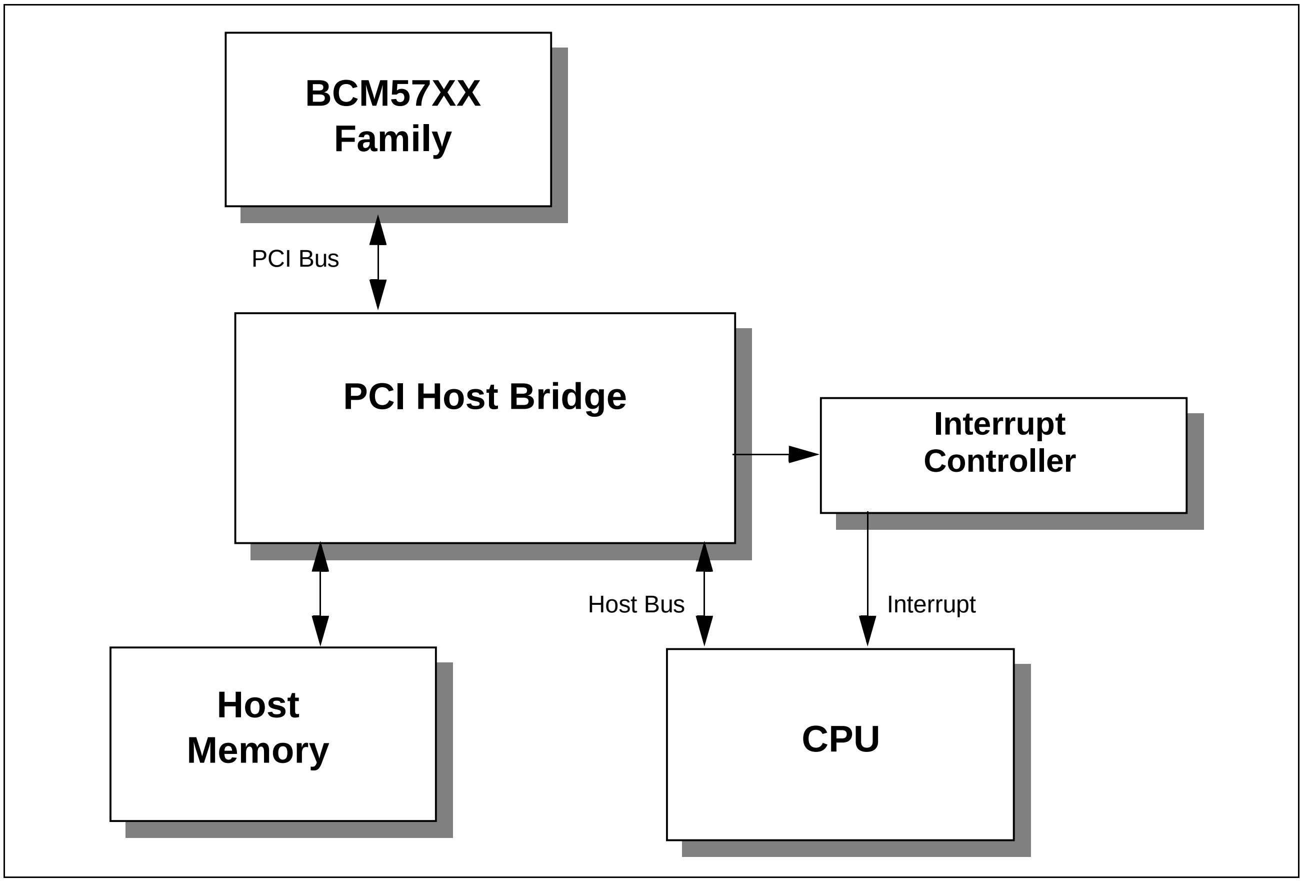


Figure 105: Message-Signaled Interrupt Scheme

Similar example in traditional interrupt scheme is used again here to illustrate MSI concept. The BCM57XX family receives one or more packets from the networks. The BCM57XX family does the following:

- DMAs data of received packets to the host.
- DMAs receive buffer descriptors to receive return ring in the host memory.
- DMAs status block to the host memory.
- Writes specified DWORD data to specified host address.

In this mode, the BCM57XX family writes DWORD data to specified host address instead of generating an interrupt. The specified data and address are configurable. The specified address is typically a memory-mapped IO port within the PCI Host Bridge. The PCI Host Bridge is the gateway to the main memory controller. This means that the DWORD data write (MSI message) to PCI Host Bridge is in the posted write buffers and was posted after the writes for the status block update. It is the rule that PCI Host Bridge must perform posted writes in the same order that they were received. This means that by the time MSI message arrives at the PCI Host Bridge, the status block has already been posted to the host memory. Upon receipt of the MSI message write, the PCI Host Bridge generates the interrupt request to the processor. Interrupt service routine of the BCM57XX device driver is invoked. It is not necessary to do a dummy read because updated status block is already in the host memory.

PCI Configuration Registers

Operation system/system software can configure the specified DWORD data and specified 64-bit host address for the device with MSI_DATA (Offset 0x64) and MSI_Address register (Offset 0x5c), respectively.

MSI Address

This is a 64-bit field. MSI address at offset 0x5c and 0x60 should be programmed with the low-order and high-order bits of the 64-bit physical address. If the Host only supports 32-bit physical address, the high-order address should be programmed with zeros.

MSI Data

This is a 16-bit field. The least significant three bits can be modified by the BCM57XX family when it writes MSI message to HOST. The DWORD data for the MSI message is depicted as shown in Figure 106.

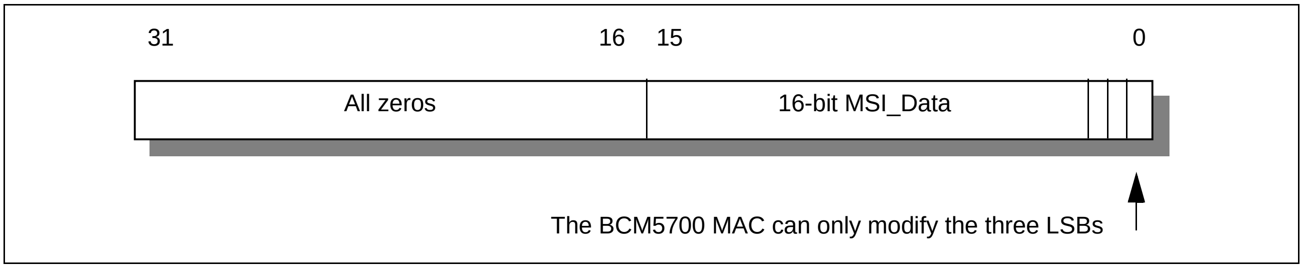


Figure 106: MSI Data FIeld

The BCM57XX family can support up to eight message types, and these MSI messages can be generated by two sources: Host Coalescing Engine or Firmware.

HOST COALESCING ENGINE

After the Host Coalescing Engine updates the status block on the host (due to receive indication, transmit completion, and so on), it either generates an interrupt or writes a MSI message if MSI is enabled. The least significant 3-bit of the MSI message originating from Host Coalescing block is configurable and can be configured by programming bits 4, 5, and 6 of the Host_Coalesing_Mode register. The default of these bits is zeros.

FIRMWARE

The BCM57XX family provides a way for firmware executed by RX RISC or TX RISC to generate MSI messages. Firmware can generate MSI messages by using MSI_FIFO_Access register (Offset 0x6008). For example, if firmware wants to generate an MSI message with least significant 3-bit as 0x2, it will write 2 to MSI_FIFO_Access register. It also needs to verify that the MSI message is written successfully by reading back MSI_FIFO_Access Overflow. If this bit is zero, then the MSI message is encoded successfully and will be sent to HOST. Otherwise, the message is not encoded.

Basic Driver Interrupt Processing Flow

Flowchart for Servicing an Interrupt

The following figure shows the basic BCM57XX driver interrupt service routine flow.

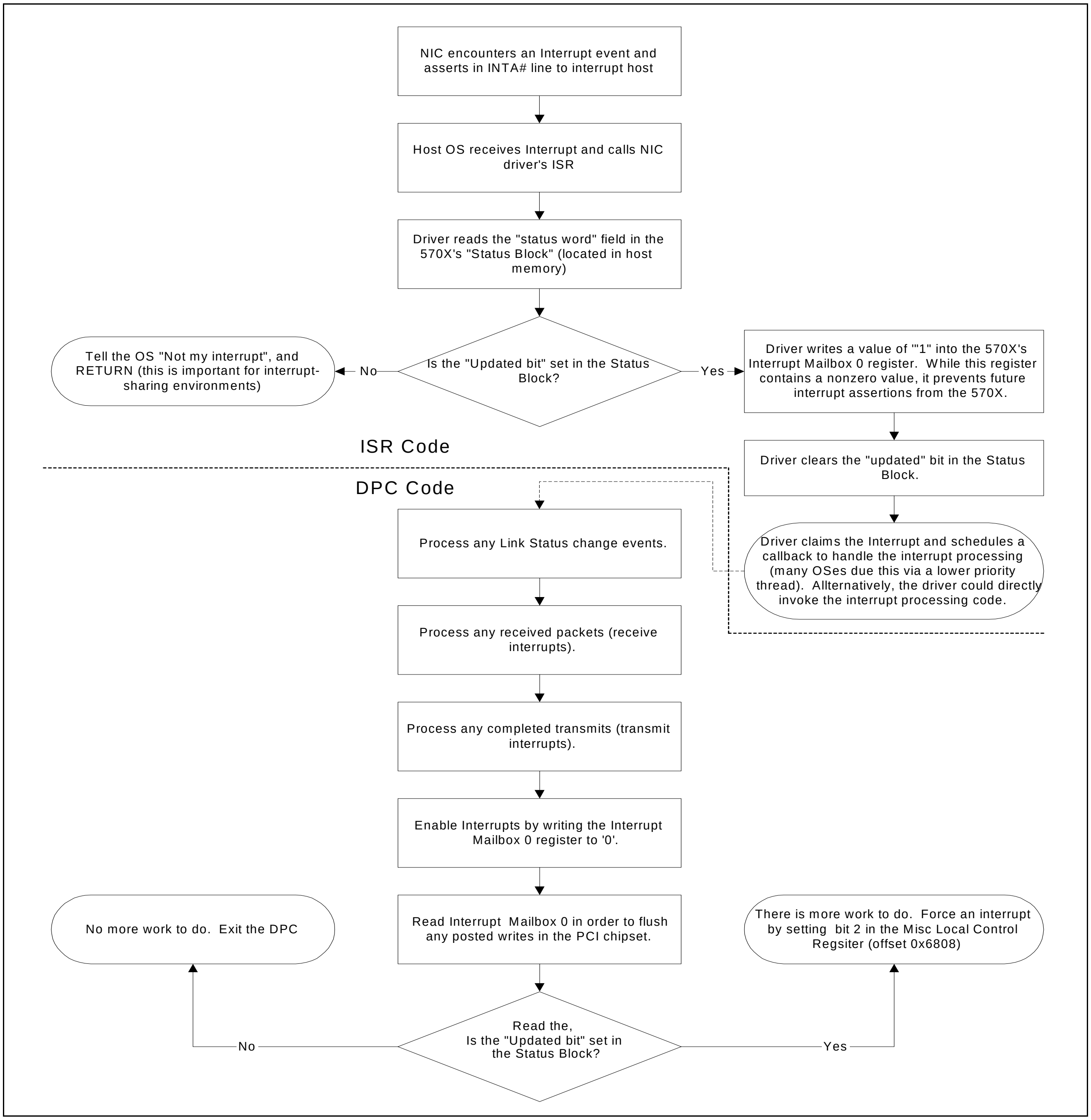

NIC encounters an Interrupt event and asserts in INTA# line to interrupt host
Host OS receives Interrupt and calls NIC driver's ISR
Driver reads the "status word" field in the 570X's "Status Block" (located in host memory)
Is the "Updated bit" set in the Status Block?
No
Tell the OS "Not my interrupt", and RETURN (this is important for interrupt-sharing environments)
Yes
Driver writes a value of '"1" into the 570X's Interrupt Mailbox 0 register. While this register contains a nonzero value, it prevents future interrupt assertions from the 570X.
Driver clears the "updated" bit in the Status Block.
Driver claims the Interrupt and schedules a callback to handle the interrupt processing (many OSes due this via a lower priority thread). Allternatively, the driver could directly invoke the interrupt processing code.
ISR Code
DPC Code
Process any Link Status change events.
Process any received packets (receive interrupts).
Process any completed transmits (transmit interrupts).
Enable Interrupts by writing the Interrupt Mailbox 0 register to '0'.
Read Interrupt Mailbox 0 in order to flush any posted writes in the PCI chipset.
Read the, Is the "Updated bit" set in the Status Block?
No
No more work to do. Exit the DPC
Yes
There is more work to do. Force an interrupt by setting bit 2 in the Misc Local Control Regsiter (offset 0x6808)

Figure 107: Basic Driver Interrupt Service Routine Flow

INTERRUPT PROCEDURE (BCM5700 ONLY)

1. Acknowledge interrupt. Write a nonzero value (i.e., value = 1) to the interrupt mailbox 0 (see "Interrupt Mailbox 0 Register (Offset 0x200)" on page 372 for host standard and flat modes and "Interrupt Mailbox 0 Register (Offset 0x5800)" on page 492 for indirect mode) to indicate that the driver is currently processing the interrupt. This step disables device interrupts except during interrupt feature.

2. Claim interrupt. Determine if BCM57XX action required. Read the Updated bit of the status word (see Table 40 on page 106). If bit is asserted, then the host coalescing engine has updated the status block.

3. Clear the Updated bit of the status word (see Table 40 on page 106). This indicates that the host driver either has or will touch the status block. If a during interrupt event is driven, the host driver can examine the Updated bit to determine if a fresh status block has been moved to host memory space.

4. Check for RX traffic.

- Loop through enabled RX Return Rings (1 to 16).
- Check for difference between RX Return Ring Producer index (Status block) and RX Return Ring Consumer index (value written to mailbox on previous call) are the number of frames to process for RX Return Ring.
- Process the packet.
- Update the RX Return Ring consumer pointer in each mailbox for new RX frames.

5. Check for TX completes.

- Loop through enabled TX Send Rings.
- Check for difference between previous consumer index (software kept) and current consumer index in the status block. These are the TX BDs which can be made available to next send operation.
- Update the previous consumer index (i.e., next call) to the value of the status block consumer index.

6. Check the Error bit in status word (optional, see Table 40 on page 106). The driver may check the state machine/FTQ status registers for various attentions.

> **Note:** Broadcom bootcode may place the RX and TX RISCs in a halted state upon firmware completion. The halted state will assert the error bit, since a RXCP_ATTN and TXCP_ATTN is generated. The host device driver will usually read the asserted Error bit, during the normal operation of the controller.

7 Enable interrupts. Write a zero value (i.e., value = 0) to the interrupt mailbox 0 (see "Interrupt Mailbox 0 Register (Offset 0x200)" on page 372 for host standard and flat modes and "Interrupt Mailbox 0 Register (Offset 0x5800)" on page 492 for indirect mode) to indicate that the ISR is done processing RX/TX. If running in indirect mode, it is also necessary to set the Clear Interrupt bit.

8 Flush status block (i.e. force update of status blocks cached by PCI bridge).

- Read interrupt mailbox (see "Interrupt Mailbox 0 Register (Offset 0x200)" on page 372 for host standard and flat modes and "Interrupt Mailbox 0 Register (Offset 0x5800)" on page 492 for indirect mode).
- Check the Updated bit in the status word (see Table 40 on page 106) located in the status block. If the Updated bit is asserted, force a new interrupt by setting Set Interrupt bit of the Miscellaneous Local Control register (see "Miscellaneous Local Control Register (Offset 0x6808)" on page 507).

INTERRUPT PROCEDURE (BCM5701 AND LATER)

1. Acknowledge interrupt. Write a nonzero value (i.e., value = 1) to the interrupt mailbox 0 (see "Interrupt Mailbox 0 Register (Offset 0x200)" on page 372 for host standard and flat modes and "Interrupt Mailbox 0 Register (Offset 0x5800)" on page 492 for indirect mode) to indicate that the driver is currently processing the interrupt. This step disables device interrupts except during interrupt feature.
2. Read and save the value of the Status Tag field of the Status Block (see Table 39 on page 105).
3. Claim interrupt. Determine if BCM57XX action required. Read the Updated bit of the status word (see Table 40 on page 106). If the Updated bit is asserted, then the host coalescing engine has updated the status block.
4. Clear the Updated bit of the status word (see Table 40 on page 106). This indicates that the host driver either has or will touch the status block. If a during interrupt event is driven, the host driver can examine the Updated bit to determine if a fresh status block has been moved to host memory space.
5. Check for RX traffic.
   - Loop through enabled RX Return Rings (1 to 16).
   - Check for difference between RX Return Ring Producer index (Status block) and RX Return Ring Consumer index (value written to mailbox on previous call) are the number of frames to process for RX Return Ring.
   - Process the packet.
   - Update the RX Return Ring consumer pointer in each mailbox for new RX frames.
6. Check for TX completes.
   - Loop through enabled TX Send Rings.
   - Check for difference between previous consumer index (software kept) and current consumer index in the status block. These are the TX BDs which can be made available to next send operation.
   - Update the previous consumer index (i.e., next call) to the value of the status block consumer index.
7. Compare the current value of the Status Tag to the saved value of the Status Tag. Flush status block (i.e., force update of status blocks cached by PCI bridge).
   - Read interrupt mailbox (see "Interrupt Mailbox 0 Register (Offset 0x200)" on page 372 for host standard and flat modes and "Interrupt Mailbox 0 Register (Offset 0x5800)" on page 492 for indirect mode).
   - Check the Updated bit in the status word (see Table 40 on page 106) located in the status block. If the Updated bit is asserted, then new data has been DMAed to the host. Repeat steps 5 and 6.
8. Check the Error bit in status word (optional, see Table 40 on page 106). The driver may check the state machine/FTQ status registers for various attentions.
9. Enable interrupts. Write the saved Status Tag to the upper 8 bits of Interrupt Mailbox 0 (see "Interrupt Mailbox 0 Register (Offset 0x200)" on page 372 for host standard and flat modes and "Interrupt Mailbox 0 Register (Offset 0x5800)" on page 492 for indirect mode), and 0 to the remaining bits (23 down to 0) to indicate that the ISR is done processing RX/TX. This also clears existing interrupts.

Interrupt Processing (Not Applicable to BCM5700)

The BCM5701 implements legacy support of the BCM5700 revision C0 interrupt mode plus five other optional modes. Each mode can be enabled individually.

Broadcom Mask Mode

Enabled by setting the Mask_Interrupt_Mode bit (bit 8) of the Miscellaneous Host Control register (see "Miscellaneous Host Control Register (Offset 0x68)" on page 325). When enabled, setting the mask bit of the Miscellaneous Host Control register will mask (de-assert) the INTA_L signal at the pin, but it will not clear the interrupt state and it will not latch the INTA_L value. Clearing the mask bit will enable the interrupt state to propagate to the INTA_L signal. When the Mask_Interrupt_Mode bit is cleared (by default), the Mask_Interrupt_Mode bit will behave in the same manner as the BCM5700 revision C0.

It should be noted that for the BCM5700 MAC, when the Mailbox 0 is nonzero or the Mask Interrupt bit is set, the During Interrupt coalescing counters and timers are used for interrupt generation; but for the rest of the BCM57XX family, the During Interrupt counters are only used when the Mailbox 0 is set.

Broadcom Tagged Status Mode

Enabled by setting the Status Tagged Status Mode bit of the Miscellaneous Host Control register (see "Miscellaneous Host Control Register (Offset 0x68)" on page 325). When enabled, a unique eight-bit tag value will be inserted into the Status Block Status Tag at location 7:0. The Status Tag can be returned to the Mailbox 0 register at location 31:24 by the host driver. When the Mailbox 0 register field 23:0 is written with a zero value, the tag field of the Mailbox 0 register is compared with the tag field of the last Status Block to be DMA'd to the host. If the tag returned is not equivalent to the tag of the first Status Block DMA'd, the interrupt status is entered.

Clear Ticks on BD Events Mode

Enabled by setting the Clear Ticks Mode on RX or the Clear Ticks Mode on TX bits of the Host Coalescing Mode register (see "Host Coalescing Mode Register (Offset 0x3C00)" on page 452). When enabled, the counters initialize to the idle state and begin counting only after a receive or transmit BD event is detected. The Receive Host Coalescing Tick counter ("Receive Coalescing Ticks Registers (Offset 0x3C08)" on page 453) is reset when a receive BD event has caused the status block update, and the Send Host Coalescing Tick counter (see "Send Coalescing Ticks Register (Offset 0x3C0C)" on page 453) is reset when a send BD event has caused a status block update. Additionally, the Host Coalescing Ticks counters are reset when a buffer descriptor has been handled by the send data initiator state machine on the send side, or the receive data completion state machine has returned a buffer descriptor on the receive side for the send and receive ticks counters respectively. When disabled, the ticks counters behave as defined for the BCM5700 MAC.

No Interrupt on Force Update

Enabled by setting the No Interrupt on Force bit of the Host Coalescing Mode register (see "Host Coalescing Mode Register (Offset 0x3C00)" on page 452). When enabled, writing the Force update bit of the Host Coalescing Mode register will cause a status update without a corresponding interrupt event.

NO INTERRUPT ON DMAD FORCE

Enabled by setting the No Interrupt on DMAD force bit of the Host Coalescing Mode register (see "Host Coalescing Mode Register (Offset 0x3C00)" on page 452). When enabled, the BD_FLAG_COAL_NOW bit of the buffer descriptor may be set to force a status block update without a corresponding interrupt.

Section 12: Register Definitions

Note: All registers and bit fields in this section are applicable to all NetXtreme devices described in this document unless otherwise noted.

Note: In the register description tables, the following notations are used in the Access column to describe the register read/write access capabilities.

- R/W = read/write
- RO = read only
- LH = latches high value (until read)
- LL = latches low value (until read)
- H = forced high
- L = forced low
- SC = self-clearing
- CR = clear on read

PCI Configuration Registers

The following describes the registers required for configuration by the PCI, PCI-X, and PCIe specifications. Access to these registers can be obtained through either the PCI Configuration address space, or through the shared-memory region of the BCM57XX family. Some registers must be defined as read-only in the PCI Configuration address space, but are allowed to be read/write when accessed by other means. A more detailed description of each register can be obtained from both the PCI, PCI-X, and PCIe specifications. All reserved fields in the Configuration region return a 0 value on a read operation. A write operation has no effect.

Table 134: PCI Configuration Register Summary

Offset	*Register*
0x00-0x01	Vendor ID
0x02-0x03	Device ID
0x04-0x05	Command
0x06-0x07	Status
0x08	Revision ID
0x09-0x0b	Class Code
0x0c	Cache Line Size
0x0d	Latency Timer
0x0e	Header Type
0x0f	BIST
0x10-0x13	Base Address Reg 1 (lower 32-bit)
0x14-0x17	Base Address Reg 2 (upper 32-bit)
0x18-0x1B	MAC 0 XBAR

Table 134: PCI Configuration Register Summary (Cont.)

Offset	Register
0x1C-0x27	Base Address Reg 4-6 (not supported)
0x2c-0x2d	Subsystem Vendor ID
0x2e-0x2f	Subsystem ID
0x30-0x33	Expansion ROM Base Address
0x34	Capabilities Pointer
0x35-0x3b	Reserved
0x3c	Interrupt Line
0x3d	Interrupt Pin
0x3e	MIN_GNT
0x3f	MAX_LAT
0x40	PCI-X Capabilities
0x41	Next Capability Pointer (PM)
0x42-0x43	PCI-X Command
0x44-0x47	PCI-X Status
0x48	Power Management Capability ID
0x49	Next Capability Pointer (VPD)
0x4a-0x4b	Power Management Capabilities
0x4c-0x4d	Power Management Control/Status
0x4e	Reserved
0x4f	Power Management Data
0x50	Vital Product Data Capability ID
0x51	Next Capability Pointer (MSI)
0x52-0x53	VPD Address/Flag
0x54-0x57	VPD Data
0x58	MSI Capability ID
0x59	Next Capability Pointer (null)
0x5a-0x5b	MSI Control
0x5c-0x63	MSI Address (64-bit)
0x64-0x65	MSI Data
0x66-0x67	Hardware Fix register
0x68-0x6b	Miscellaneous Host Control
0x6c-0x6f	DMA Read/Write Control
0x70-0x73	PCI State
0x74-0x77	PCI Clock Control
0x78-0x7b	Register Base Address
0x7c-0x7f	Memory Window Base Address
0x80-0x83	Register Data [1]
0x84-0x87	Memory Window Data
0x88-0x8b	Mode Control

Table 134: PCI Configuration Register Summary (Cont.)

Offset	Register
0x8c-0x8f	Miscellaneous Configuration
0x90-0x93	Miscellaneous Local Control
0x94-0x97	Reserved
0x98-0x9f	UNDI Receive BD Standard Ring Producer Index Mailbox
0xa0-0xa7	UNDI Receive Return Ring Consumer Index Mailbox
0xa8-0xaf	UNDI Send BD Producer Index Mailbox
0xb0-0xb7	Interrupt Mailbox 0 (shadow, see "Interrupt Mailbox 0 Register (Offset 0x200)" on page 372 for host standard and flat modes and "Interrupt Mailbox 0 Register (Offset 0x5800)" on page 492 for indirect mode)
0xb8-0xbb	Dual MAC control register
0xbc-0xbf	MAC Message Exchange Output register
0xc0-0xc3	MAC Message Exchange Input register
0xc4-0xcf	Reserved
0xd0	PCIe Capability ID (for BCM5721, BCM5751, and BCM5752 only)
0xd1	Next Capability Pointer (for BCM5721, BCM5751, and BCM5752 only)
0xd2-0xd3	PCIe Capabilities (for BCM5721, BCM5751, and BCM5752 only)
0xd4-0xd7	Device Capabilities (for BCM5721, BCM5751, and BCM5752 only)
0xd8-0xd9	Device Control (for BCM5721, BCM5751, and BCM5752 only)
0xda-0xdb	Device Status (for BCM5721, BCM5751, and BCM5752 only)
0xdc-0xdf	Link Capabilities (for BCM5721, BCM5751, and BCM5752 only)
0xe0-0xe1	Link Control (for BCM5721, BCM5751, and BCM5752 only)
0xe2-0xe3	Link Status (for BCM5721, BCM5751, and BCM5752 only)
0xe4-0xff	Reserved

[1] These registers are remapped by the PCI Block with the associated address upon host access. The CPU can only access these registers directly through the Control registers.

VENDOR ID REGISTER (OFFSET 0X00)

The 16-bit Vendor ID register identifies the PCI adapter manufacturer. Valid vendor identifiers are allocated by the PCI SIG to ensure uniqueness. This register defaults to 0x12AEh at power-on reset for BCM5700. This register may be written by the internal RISCs. Normally, this register is loaded with the appropriate vendor ID after reset by firmware that resides in the NVRAM. Broadcom's Vendor ID is 0x14E4.

Table 135: Vendor ID Register (Offset 0x00)

Bit	Field	Description	Init	Access
15-0	Vendor ID	Reflects manufacturer of the part. Reflects manufacturer of the part (BCM5700 only).	0x14E4[a] 0x12AE[a]	R/W[b] R/W[b]
		Reflects manufacturer of the part (BCM5788 only).	0x173B[a]	R/W[b]

[a] Value shown is after hardware reset.
[b] Not writable by PCI configuration access.

Device ID Register (Offset 0x02)

The 16-bit Device ID register identifies the particular adapter within those made by the same manufacturer. This register defaults to the value specified in Table 136 at power-on reset but will be modified by the bootcode firmware to match the values in Table 2: "Family Revision Levels," on page 5. This register may be written by the internal RISCs.

Table 136: Device ID Register (Offset 0x02)

Bit	Field	Description	Init	Access
15-0	Device ID	Unique identifier for the BCM5700 MAC	0x0003[1]	R/W[2]
		Unique identifier for the BCM5701 MAC Transceiver	0x1645[1]	R/W[2]
		Unique identifier for the BCM5702 MAC Transceiver	0x16A6[1]	R/W[2]
		Unique identifier for the BCM5703C MAC Transceiver	0x16A7[1]	R/W[2]
		Unique identifier for the BCM5703S MAC Transceiver SerDes	0x16A7[1]	R/W[2]
		Unique identifier for the BCM5704C Dual-MAC Transceiver	0x1648[1]	R/W[2]
		Unique identifier for the BCM5704S Dual-MAC Transceiver SerDes	0x1649[1]	R/W[2]
		Unique identifier for the BCM5705M MAC Transceiver	0x165d[1]	R/W[2]
		Unique identifier for the BCM5705 MAC Transceiver	0x1653[1]	R/W[2]
		Unique identifier for the BCM5788 MAC Transceiver	0x03ED[1]	R/W[2]
		Unique identifier for the BCM5721 MAC Transceiver	0x1677[1]	R/W[2]
		Unique identifier for the BCM5751 MAC Transceiver	0x1677[1]	R/W[2]
		Unique identifier for the BCM5751M MAC Transceiver	0x167d [1]	R/W[2]
		Unique identifier for the BCM5752 MAC Transceiver	0x1600[1]	R/W[2]
		Unique identifier for the BCM5752M MAC Transceiver	0x1601[1]	R/W[2]
		Unique identifier for the BCM5714C Dual-MAC Transceiver	0x1668[1]	R/W[2]
		Unique identifier for the BCM5714S Dual-MAC Transceiver SerDes	0x1669[1]	R/W[2]
		Unique identifier for the BCM5715C Dual-MAC Transceiver	0x1678[1]	R/W[2]
		Unique identifier for the BCM5715S Dual-MAC Transceiver SerDes	0x1679[1]	R/W[2]

[1] Value shown is after hardware reset.
[2] Not writable by PCI configuration access.

Command Register (Offset 0x04)

The 16-bit Command register is used by the PCI-based host to enable various features of the device. All of the bit positions are predefined by the PCI specification. Not all bits in this register are implemented.

Table 137: Command Register (Offset 0x04)

Bit	Field	Description	Init	Access
15-11	Reserved	Reserved.	0	R/O
10	Interrupt Disable (BCM5703 B0 or later, BCM5704 B0 or later, BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	Setting this bit to 1 disables the device from asserting INTA on the PCI bus. This bit does not affect the internal state of the INTA request.	0	R/W
	Reserved	Reserved.	0	R/O
9	Fast Back-to-Back Enable	Enables fast back-to-back transactions to different devices.This device does not support this capability, therefore, this bit is hardwired to 0.	0	R/O
8	System Error Enable	Enables system error detection. The device reports address parity errors when this bit is set if parity error detection is enabled.	0	R/W
7	Stepping Control	Controls whether address/data stepping is done. This device does not do stepping, therefore, this bit is hardwired to 0.	0	R/O
6	Parity Error Enable	Enables data parity error detection. The device reports data parity errors when this bit is set.	0	R/W
5	VGA Palette Snoop	Enables palette snoop on VGA devices. This device does not support this capability, therefore, this bit is hardwired to 0.	0	R/O
4	Memory Write and Invalidate	The BCM57XX family does not support the MWI command, therefore, this bit should remain cleared to 0.	0	R/O (BCM5721, BCM5751, and BCM5752 only) R/W (others)
3	Special Cycles	Enables device to monitor Special Cycles operations. This device does not support Special Cycles, therefore, this bit is hardwired to 0.	0	R/O
2	Bus Master	Enables bus mastering. The device will not act as a bus master until this bit is set. ***Note:*** In PCI-X mode, the device is permitted to initiate a split completion regardless of the state of this bit.	0	R/W
1	Memory Space	Enables Memory space accesses. The device will not respond to Memory accesses until this bit is set.	0	R/W
0	I/O Space	Enables I/O space accesses. This device does not support I/O space, therefore, this bit is hardwired to 0.	0	R/O

STATUS REGISTER (OFFSET 0X06)

The 16-bit Status register is used to indicate status information to the PCI-based host for PCI bus-related events. All of the bit positions are predefined by the PCI specification. Not all bits in this register are implemented.

Table 138: Status Register (Offset 0x06)

Bit	Field	Description	Init	Access
15	Detected Parity Error	Indicates a data parity error was detected even if parity reporting was not enabled.	0	R/W2C
14	Signaled System Error	Indicates this device asserted system error ($\overline{\text{SERR}}$).	0	R/W2C
13	Received Master Abort	Indicates this device was a bus master and the transaction was terminated with a master-abort.	0	R/W2C
12	Received Target Abort	Indicates this device was a bus master and received a target-abort.	0	R/W2C
11	Signaled Target Abort	Indicates this device initiated a target-abort. This bit is only set if an external master disappears during a target operation.	0	R/W2C
10-9	DEVSEL Timing	These bits encode the slowest timing of $\overline{\text{DEVSEL}}$, except for configuration cycles. Valid entries are 00 for fast, 01 for medium, and 10 for slow. The device is capable of fast timing and these two bits are hardwired to 00.	00 (BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	R/O
		These bits encode the slowest timing of $\overline{\text{DEVSEL}}$, except for configuration cycles. Valid entries are 00 for fast, 01 for medium, and 10 for slow. The device is capable of medium timing and these two bits are hardwired to 01.	01	R/O
8	Master Data Parity Error	Indicates that this device was a bus master when a parity error was detected and reporting of parity errors is enabled. The device is capable of operating with this bit set.	0	R/W2C
7	Fast Back-to-Back Capable	Indicates whether fast back-to-back transactions can be accepted when transactions are not to the same agent.	0 (BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	R/O
		Indicates whether fast back-to-back transactions can be accepted when transactions are not to the same agent. The device is capable of accepting fast back-to-back transactions, so this bit is hardwired to 1.	1	R/O
6	Reserved		0	R/O

Table 138: Status Register (Offset 0x06) (Cont.)

Bit	Field	Description	Init	Access
5	66 MHz Capable	Indicates whether this device can operate on a 66-MHz PCI bus.	0 (BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	R/O
		Indicates whether this device can operate on a 66-MHz PCI bus. The device is 66 MHz capable, so this bit is hardwired to 1.	1 (other devices)	R/O
4	Capabilities List	Indicates whether this device has a capabilities list. The device has a capabilities list, so this bit is hardwired to 1.	1	R/O
3	Interrupt Status (BCM5704 B0 and later, BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	Indicates this device has generated an interrupt.	0	R/O
	Reserved (other devices)		0	R/O
2-0		Reserved.	0	R/O

To clear a particular bit position, a write must be performed with that bit being a 1 and all other bit positions being a 0. In this manner, bits can be set by hardware conditions and cleared by software.

Note: On a dual port devices such as the BCM5704, bits 8 and 11 to 15 of the Status register (offset 0x06) are shared between the two PCI functions. This means that if the Detected Parity Error bit is set on PCI function 0, it is also set on PCI function 1.

Revision ID Register (Offset 0x08)

The 8-bit Revision ID register is used by the manufacturer to identify the specific revision number of this adapter. Any value is allowable. It is recommended that this field be initialized to the board revision level. This register defaults to 0x00 at power-on reset. This register may be written by the internal RISCs but cannot be written via the host PCI interface. Normally, this register is loaded with the appropriate Revision ID after reset by firmware that resides in the NVRAM.

Note: See the latest errata documentation for the latest part revision codes (see “Revision Levels” on page 5).

Table 139: Revision ID Register (Offset 0x08)

Bit	Field	Description	Init	Access
8-0	Revision ID	Revision of this part.	0h	R/W[1]

[1] Not writable by PCI configuration access.

Class Code Register (Offset 0x09)

The 24-bit Class Code register identifies the generic function of the device. All of the legal values are specific in the PCI specification. This field is hardwired to the class code for an Ethernet interface (0x020000).

Table 140: Class Code Register (Offset 0x09)

Bit	Field	Description	Init	Access
23-0	Class Code	PCI class code for this device.	020000h	R/O

Cache Line Size Register (Offset 0x0C)

The 8-bit Cache Line Size register is used by the PCI-based host to indicate the size of a cache line in 32-bit increments. This field affects how the Memory Write and Invalidate command is utilized. If this register is zero, then the Memory Write and Invalidate command will not be used by the device. This register may be written via either the host PCI interface or the internal RISCs. Normally, this register will be set by the host PCI BIOS during the boot up sequence. This register defaults to 0. The BCM57XX family supports cache line sizes of 8, 16, 32, 64, 128, and 512 bytes.

This register does not apply in PCIe systems and is implemented as a read-write field for legacy compatibility purposes only. It has no impact on any PCIe device functionality.

Table 141: Cache Line Size Register (Offset 0x0C)

Bit	Field	Description	Init	Access
7-0	Reserved (BCM5721, BCM5751, and BCM5752 only)		0	R/W
	Cache line size (other devices)	System cache line size.	0	R/W

Latency Timer Register (Offset 0x0D)

The 8-bit Latency Timer register is used by the PCI-based host to indicate the number of PCI clocks in which the device may own the bus before checking to see if the bus should be relinquished. The device will relinquish the bus if the latency timer expires and $\overline{\text{GNT}}$ has been revoked. Only the upper 5-bits are usable as the lower 3 bits are hardwired to 0. This register may be written via either the host PCI interface or the internal RISCs. Normally, this register will be set by the host PCI BIOS during the boot up sequence. After a reset, this register defaults to 0x0 or 0x40 for conventional PCI or PCI-X respectively. This register does not apply in PCIe.

Table 142: Latency Timer Register (Offset 0x0D)

Bit	Field	Description	Init	Access
7-0	Reserved (BCM5721, BCM5751, and BCM5752 only)		0000h	R/O
	Latency Timer (other devices)	Number of PCI clocks in which the device may own the bus before checking to see if the bus should be relinquished.	0000h (PCI) 0040h (PCI-X)	R/W

Header Type Register (Offset 0x0E)

The 8-bit Header Type register identifies the layout of bytes 10h through 3Fh 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.

Table 143: Header Type Register (Offset 0x0E)

Bit	Field	Description	Init	Access
7-0	Header Type	Identifies this device as having a single function.	0	R/O

BIST Register (Offset 0x0F)

The 8-bit BIST register is used to initiate and report the results of any Built-In Self-Test. This device does not export BIST results to this register. Therefore, this register defaults to 0x00 at power-on reset. Optionally, firmware could be developed to execute a self-test and write the result into this register because this register may be written by the internal RISCs. This register cannot be written via the host PCI interface.

Table 144: BIST Register (Offset 0x0F)

Bit	Field	Description	Init	Access
7-0	BIST	Built-in self-test.	0	R/W[1]

[1] Not writable by PCI configuration access.

Base Address Register 1/2 Register (Offset 0x10-0x17)

The 64-bit Base Address register is used to establish the memory space that the adapter requires within the system. Once the system has determined the needs of all of the adapters, the operating system can be booted.

The device supports one 64-bit Base Address register which must be located in the host's memory space. The other four 32-bit Base Address registers are not implemented and will each return 0x0000 when read.

Base Address register 1/2 is required for the device to function. It provides either a 64 KB or a 32 MB control region from which the host can access the adapter. The size of the region is determined by the selected host View (bit 8 of PCI State register).

Table 145: Base Address Register 1/2 (Offset 0x10)

Bit	Field	Description	Init	Access
63-32	Extended Base Address	High order address bits.	X	R/W
31-xx+1	Base Address	Low order address bits.	0	R/W
xx-4	Size Indication	Portion of the address bits that are used to indicate the size of the PCI address map. These are all set to 0. *xx* is equal to 15 for Standard View, and to 24 for Flat View.	0	R/O
3	Prefetchable	Indicates that there are no side effects on reads; the device returns all bytes regardless of byte enables, and processor writes can get merged. This bit is always 0.	0	R/O

Table 145: Base Address Register 1/2 (Offset 0x10) (Cont.)

Bit	*Field*	*Description*	*Init*	*Access*
2-1	Type	Encoded with the following values: 00: Located anywhere in 32-bit address space 01: Reserved 10: Located anywhere in 64-bit address space 11: Reserved The device is capable of being located anywhere within the 64-bit address space, so these bits are hard-coded to 10.	10	R/O
0	Memory Space Indicator	This bit is always 0. Base Address registers map to Memory Space.	0	R/O

MAC 0 XBAR Register (Offset 0x18)

This register is applicable to BCM5704C, BCM5704S, BCM5714C, BCM5714S, BCM5715C, and BCM5715S devices only.

Table 146: MAC 0 XBAR Register (Offset 0x18)

Bit	*Field*	*Description*	*Init*	*Access*
31-0	XBAR value	This register is used to allow the Host to access the MAC1's memory/register space via memory Write/Read command. The content of this register is not cleared by software or hardware reset. Once programmed, the value remains the same until the next software write.	X	R/W

MAC 0 XBAR Register (Upper) (Offset 0X1C)

This register is applicable to BCM5704C, BCM5704S, BCM5714C, BCM5714S, BCM5715C, and BCM5715S devices only.

Table 147: MAC 0 XBAR Register (Upper) (Offset 0x1C)

Bit	*Field*	*Description*	*Init*	*Access*
31-0	XBAR value (Upper 32 bits)	This upper 32-bits of BAR to access the MAC 1's memory/register space.	0x0000	R/W

Subsystem Vendor ID Register (Offset 0x2C)

The 16-bit Subsystem Vendor ID register is used by the board manufacturer for identification. This register may differ from the Vendor ID if the board manufacturer is different than the chip manufacturer. This register defaults to 0x12AE at power-on reset. This register may be written by the internal RISCs. Normally, this register is loaded with the appropriate Subsystem Vendor ID after reset by firmware that resides in the NVRAM and its value varies by board OEM.

Table 148: Subsystem Vendor ID Register (Offset 0x2C)

Bit	Field	Description	Init	Access
15-0	Subvendor ID	Identifies board manufacturer (BCM5700 MAC)	0x12AE[1]	R/W[2]
		Identifies board manufacturer (BCM5701 MAC Transceiver)	0x14E4[1]	R/W[2]
		Identifies board manufacturer (BCM5702 MAC Transceiver)	0x14E4[1]	R/W[2]
		Identifies board manufacturer (BCM5703C MAC Transceiver)	0x14E4[1]	R/W[2]
		Identifies board manufacturer (BCM5703S MAC Transceiver SerDes)	0x14E4[1]	R/W[2]
		Identifies board manufacturer (BCM5704C Dual-MAC Transceiver)	0x14E4[1]	R/W[2]
		Identifies board manufacturer (BCM5704S Dual-MAC Transceiver SerDes)	0x14E4[1]	R/W[2]
		Identifies board manufacturer (BCM5705 MAC Transceiver)	0x14E4[1]	R/W[2]
		Identifies board manufacturer (BCM5705M MAC Transceiver)	0x14E4[1]	R/W[2]
		Identifies board manufacturer (BCM5788 MAC Transceiver)	0x173B[1]	R/W[2]
		Identifies board manufacturer (BCM5721 MAC Transceiver)	0x14E4[1]	R/W[2]
		Identifies board manufacturer (BCM5751 MAC Transceiver)	0x14E4[1]	R/W[2]
		Identifies board manufacturer (BCM5752 MAC Transceiver)	0x14E4[1]	R/W[2]
		Identifies board manufacturer (BCM5752M MAC Transceiver)	0x14E4[1]	R/W[2]
		Identifies board manufacturer (BCM5714C Dual-MAC Transceiver)	0x14E4[1]	R/W[2]
		Identifies board manufacturer (BCM5714S Dual-MAC Transceiver)	0x14E4[1]	R/W[2]
		Identifies board manufacturer (BCM5715C Dual-MAC Transceiver)	0x14E4[1]	R/W[2]
		Identifies board manufacturer (BCM5715S Dual-MAC Transceiver)	0x14E4[1]	R/W[2]

[1] Value shown is after hardware reset.

[2] Not writable by PCI configuration access.

SUBSYSTEM ID REGISTER (OFFSET 0X2E)

The 16-bit Subsystem ID register is used by the board manufacturer for identification. This register may be used to differentiate between different boards that use the same PCI silicon component. This register may be written by the internal RISCs. Normally, this register is loaded with the appropriate Subsystem Device ID after reset by boot code firmware and its value varies by board OEM and NIC/LOM.

Table 149: Subsystem ID Register (Offset 0x2E)

Bit	*Field*	*Description*	*Init*	*Access*
15-0	Subsystem ID	ID assigned by board manufacturer (BCM5700 MAC)	0x0003[1]	R/W[2]
		ID assigned by board manufacturer (BCM5701 MAC Transceiver)	0x1645[1]	R/W[2]
		ID assigned by board manufacturer (BCM5702 MAC Transceiver)	0x16A6[1]	R/W[2]
		ID assigned by board manufacturer (BCM5703C MAC Transceiver)	0x16A7[1]	R/W[2]
		ID assigned by board manufacturer (BCM5703S MAC Transceiver SerDes)	0x16A7[1]	R/W[2]
		ID assigned by board manufacturer (BCM5704C Dual-MAC Transceiver)	0x1648[1]	R/W[2]
		ID assigned by board manufacturer (BCM5704S Dual-MAC Transceiver SerDes)	0x1648[1]	R/W[2]
		ID assigned by board manufacturer (BCM5705 MAC Transceiver)	0x1653[1]	R/W[2]
		ID assigned by board manufacturer (BCM5705M MAC Transceiver)	0x165d[1]	R/W[2]
		ID assigned by board manufacturer (BCM5788 MAC Transceiver)	0x03ED[1]	R/W[2]
		ID assigned by board manufacturer (BCM5721 MAC Transceiver)	0x1659[1]	R/W[2]
		ID assigned by board manufacturer (BCM5751 MAC Transceiver)	0x1677[1]	R/W[2]
		ID assigned by board manufacturer (BCM5751M MAC Transceiver)	0x167d[1]	R/W[2]
		ID assigned by board manufacturer (BCM5752 MAC Transceiver)	0x1600[1]	R/W[2]
		ID assigned by board manufacturer (BCM5752M MAC Transceiver)	0x1601[1]	R/W[2]
		ID assigned by board manufacturer (BCM5714C MAC Transceiver)	0x1668[1]	R/W[2]
		ID assigned by board manufacturer (BCM5714S MAC Transceiver)	0x1669[1]	R/W[2]
		ID assigned by board manufacturer (BCM5715C MAC Transceiver)	0x1678[1]	R/W[2]
		ID assigned by board manufacturer (BCM5715S MAC Transceiver)	0x1679[1]	R/W[2]

[1] Value shown is after hardware reset.
[2] Not writable by PCI configuration access.

Expansion ROM Base Address Register (Offset 0x30)

The 32-bit Expansion ROM Base Address register is used to establish the location of a 64 KB ROM region within the device's memory space. This ROM region is used for PXE support.

Table 150: Expansion ROM Base Address Register (Offset 0x30)

Bit	Field	Comments	Init	Access
31-16	ROM Base Address	Address bits.	X	R/O[1]
15-11	ROM size indication	Hardwired to 0 for 64 KB.	00000	R/O[1]
10-1	Reserved		000h	R/O
0	Expansion ROM Enable	Set to a 1 to enable the use of this ROM region (firmware only).	0	R/O[1]

[1] R/O unless expansion ROM bit is enabled.

Capabilities Pointer Register (Offset 0x34)

The 8-bit Capabilities Pointer register specifies an offset in the PCI address space of a linked list of new capabilities. The capabilities are PCI-X, PCI Power Management, Vital Product Data (VPD), Message Signaled Interrupts (MSI), and PCIe (PCIe).

Table 151: Capabilities Pointer Register (Offset 0x34)

Bit	Field	Description	Init	Access
7-0	Capabilities Pointer (BCM5705, BCM5788, BCM5721, BCM5751, and BCM5752 only)	Points to a linked list of new PCI capabilities.	48h	R/O
	Capabilities pointer (other devices)	Points to a linked list of new PCI capabilities.	40h	R/O

Interrupt Line Register (Offset 0x3C)

The 8-bit Interrupt Line register is used to communicate interrupt line routing information. This field is set after configuration by the host and later used by any driver which needs to know which physical interrupt on the system interrupt controller is assigned to this device. The device supports any value in this field.

Table 152: Interrupt Line Register (Offset 0x3C)

Bit	Field	Description	Init	Access
7-0	Interrupt Line	Identifies interrupt routing information.	0	R/W

Interrupt Pin Register (Offset 0x3D)

The 8-bit Interrupt Pin register is used to indicate which interrupt pin the device uses. The BCM5704 device supports two PCI functions and is hardwired for INTA (0x01) on PCI function 0 and INTB (0x02) on PCI function 1. All other BCM57XX devices support only one PCI function, which is hardwired for INTA (0x01).

Minimum Grant Register (Offset 0x3E)

The 8-bit Minimum Grant register is used to indicate the device's desired minimum grant period in units of 250 nanoseconds, 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 register is set to 0x40 at reset. This register does not apply to PCIe devices.

Table 153: Minimum Grant Register (Offset 0x3E)

Bit	*Field*	*Description*	*Init*	*Access*
7-0	Reserved (BCM5721, BCM5751, and BCM5752 only)		0	R/O
	Minimum Grant (other devices)	Indicates the desired minimum grant period.	40h	R/W[1]

[1] Not writable by PCI configuration access.

Maximum Latency Register (Offset 0x3F)

The 8-bit Maximum Latency register is used to indicate the device's desired maximum time between being granted the PCI bus in units of 250 nanoseconds, assuming a PCI clock rate of 33 MHz. Devices should specify values that will allow them to most effectively use their internal resources. This register is set to 0x00 at reset. This register does not apply to PCIe devices.

Table 154: Maximum Latency Register (Offset 0x3F)

Bit	*Field*	*Description*	*Init*	*Access*
7-0	Reserved (BCM5721, BCM5751, and BCM5752 only)		0	R/O
	Max Latency	Indicates desired maximum grant latency.	0	R/W[1]

[1] Not writable by PCI configuration access.

PCI-X CAPABILITIES

The PCI-X capabilities registers (offset 0x40 to 0x47) are not applicable to the BCM5705, BCM5788, BCM5721, BCM5751, and BCM5752 devices.

PCI-X devices include new status and control registers that are located in the Capabilities List in the devices PCI Configuration Space. These PCI-X capabilities registers start at offset 0x40 of PCI Configuration Space. These registers appear in Configuration Space regardless of whether the device is operating in PCI or PCI-X mode.

PCI-X CAPABILITY ID REGISTER (OFFSET 0X40)

This 8-bit register identifies this item in the Capabilities List as a PCI-X register set. This value is hardwired to 0x07 to indicate the PCI-X capabilities set.

Table 155: PCI-X Capability ID Register (Offset 0x40)

Bit	Field	Description	Init	Access
7-0	PCI-X Capability ID	Identifies this item as PCI-X capabilities.	07h	R/O

PCI-X NEXT CAPABILITIES POINTER REGISTER (OFFSET 0X41)

This 8-bit register points to the next item in the Capabilities List. This value is hard-wired to 0x48 and points to the Power Management Register block.

Table 156: PCI-X Next Capabilities Pointer Register (Offset 0x41)

Bit	Field	Description	Init	Access
7-0	PCI-X Next Capabilities	Points to the next capabilities block which is for Power Management (PM).	48h	R/O

PCI-X COMMAND REGISTER (OFFSET 0X42)

This 16-bit register controls various modes and features of the PCI-X device.

Table 157: PCI-X Command Register (Offset 0x42)

Bit	Field	Description	Init	Access
15-7	Reserved	Reserved.	00h	R/O

Table 157: PCI-X Command Register (Offset 0x42) (Cont.)

<table>
<tr><th>Bit</th><th>Field</th><th>Description</th><th>Init</th><th>Access</th></tr>
<tr><td>6-4</td><td>Maximum Outstanding Split Transactions</td><td>Sets the maximum number of Split Transactions the device is permitted to have outstanding at one time. Software may change this value at any time. The most recent value of the register is used each time the device prepares a new Sequence. At reset, this value is set to 0 to indicate that it can have one Split Transactions outstanding when the Maximum Memory Byte Count is set to 0 (512-byte).
<table>
<tr><th>Register</th><th>Maximum Outstanding</th></tr>
<tr><td>0</td><td>1</td></tr>
<tr><td>1</td><td>2</td></tr>
<tr><td>2</td><td>3</td></tr>
<tr><td>3</td><td>4</td></tr>
<tr><td>4</td><td>8</td></tr>
<tr><td>5</td><td>12</td></tr>
<tr><td>6</td><td>16</td></tr>
<tr><td>7</td><td>32</td></tr>
</table></td><td>000</td><td>R/W</td></tr>
<tr><td>3-2</td><td>Maximum Memory Read Byte Count</td><td>Indicates the maximum byte count the device uses when initiating a Sequence with one of the burst memory read commands. At reset, these bits are set to 0.
<table>
<tr><th>Register</th><th>Maximum Byte Count</th></tr>
<tr><td>0</td><td>512</td></tr>
<tr><td>1</td><td>1024</td></tr>
<tr><td>2</td><td>2048</td></tr>
<tr><td>3</td><td>4096</td></tr>
</table>
<b><i>Note:</i></b> For BCM5703C/BCM5703S in PCI-X Mode, the DMA Read Watermark in the DMA Read/Write Control register (see "DMA Read/Write Control Register (Offset 0x6C)" on page 327) should be set to less than or equal to the Maximum Memory Read Byte Count setting. For example, If Maximum Memory Read Byte Count is set to 0 (i.e., 512), the allowable DMA Read Watermarks are 0 to 4.</td><td>00</td><td>R/W</td></tr>
<tr><td>1</td><td>Enable Relaxed Ordering</td><td>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. At reset, this bit is set to 1.</td><td>1</td><td>R/W</td></tr>
<tr><td>0</td><td>Data Parity Error Recovery Enable</td><td>When set, the device should attempt to recover from data parity errors. If this bit is 0 and the device is in PCI-X mode, the device asserts $\overline{\text{SERR}}$ (if enabled) whenever the Master Data Parity Error bit (bit 8 of Status Register) is set. At reset, this bit is set to 0.</td><td>0</td><td>R/W</td></tr>
</table>

PCI-X Status Register (Offset 0x44)

Table 158: PCI-X Status Register (Offset 0x44)

<table>
<tr><th>Bit</th><th>Field</th><th>Description</th><th>Init</th><th>Access</th></tr>
<tr><td>31-30</td><td>Reserved</td><td></td><td>00</td><td>R/O</td></tr>
<tr><td>29</td><td>Received Split Completion Error Message</td><td>Indicates that the device received a Split Completion Message with the Split Completion Error attribute set. At reset, this bit is set to 0. Writing a 1 to this bit, will clear this bit.</td><td>0</td><td>R/W2C</td></tr>
<tr><td>28-26</td><td>Designed Maximum Cumulative Read Size</td><td>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 at one time. The device hardwires this field to 0 to indicate it can have 1024 outstanding bytes.
<table>
<tr><th>Register</th><th>Maximum Outstanding Bytes</th></tr>
<tr><td>0</td><td>1 KB</td></tr>
<tr><td>1</td><td>2 KB</td></tr>
<tr><td>2</td><td>4 KB</td></tr>
<tr><td>3</td><td>8 KB</td></tr>
<tr><td>4</td><td>16 KB</td></tr>
<tr><td>5</td><td>32 KB</td></tr>
<tr><td>6</td><td>64 KB</td></tr>
<tr><td>7</td><td>128 KB</td></tr>
</table></td><td>000</td><td>R/O</td></tr>
<tr><td>25-23</td><td>Designed Maximum Outstanding Split Transactions</td><td>Indicates a number greater than or equal to the maximum number of Split Transactions the device is designed to have outstanding at one time. The device hardwires this field to 0 to indicate it can have one outstanding Split Transaction. If system configuration software were to set the value of the Maximum Outstanding Split Transactions register (in the PCI-X Command register) to a value different from this register, the device would have to use the smaller value.
<table>
<tr><th>Register</th><th>Maximum Outstanding</th></tr>
<tr><td>0</td><td>1</td></tr>
<tr><td>1</td><td>2</td></tr>
<tr><td>2</td><td>3</td></tr>
<tr><td>3</td><td>4</td></tr>
<tr><td>4</td><td>8</td></tr>
<tr><td>5</td><td>12</td></tr>
<tr><td>6</td><td>16</td></tr>
<tr><td>7</td><td>32</td></tr>
</table></td><td>000</td><td>R/O</td></tr>
</table>

Table 158: PCI-X Status Register (Offset 0x44) (Cont.)

Bit	*Field*	*Description*	*Init*	*Access*
22-21	Designed Maximum Memory Read Byte Count	Indicates the 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 commands. The device sets these bits to 00 to indicate it can support up to 512 bytes. If system configuration software was to set the value of the Maximum Memory Read Byte Count register (in the PCI-X Command register) to a value different from this register, the device would have to use the smaller value. **Register** **Maximum Byte Count** 0 512 1 1024 2 2048 3 4096	00	R/O
20	Device Complexity	This bit indicates whether this device is a simple device or a bridge device. Since the device is a simple device, this bit is hardwired to 0.	0	R/O
19	Unexpected Split Completion	This bit is set if an unexpected Split Completion with this device's Requester ID is received. At reset, this bit is set to 0.	0	R/O
18	Split Completion Discarded	This bit is set if the device discards a Split Completion because the requester would not accept it. At reset, this bit is set to 0. This bit will never be set, because the target logic never does a split response.	0	R/W2C
17	133 MHz Capable	This bit indicates that the device is capable of 133-MHz operation in PCI-X Mode. This bit is hardwired to 1.	1	R/O
16	64-bit Device	This bit indicates the size of the device's AD bus. This bit is hardwired to 1 to indicate a 64-bit wide AD bus.	1	R/O
15-8	Bus Number	This field indicates the number of the bus segment for the device containing this function (there is only one function on the device). This field is read for diagnostic purposes only. At reset this field is set to 0xff.	FFh	R/O
7-3	Device Number	This field indicates the number of the device containing this function; i.e., the number in Device Number field (AD[15:11]) of the address of a Type 0 configuration transaction that is assigned to the connection of the system hardware. This field is read for diagnostic purposes only. At reset, this field is set to 0x1f.	11111	R/O
2-0	Function Number	This field indicates the number of the function; i.e., the number in Function Number field (AD[10:08]) of the address of a Type 0 configuration transaction to which this function responds. This field is read for diagnostic purposes only. At reset, this field is set to 001.	001	R/O

PCI Power Management Capabilities

Devices that support PCI Power Management must support a block of registers that is part of the Capabilities List in PCI Configuration Space. The PCI Power Management Register Block is located at offset 0x48. The device supports the following PCI Power Management registers:

Power Management Capability ID Register (Offset 0x48)

This 8-bit register identifies this item in the Capabilities List as a PCI Power Management register set.

Table 159: Power Management Capability Register (Offset 0x48)

Bit	*Field*	*Description*	*Init*	*Access*
7-0	PM Capability ID	Identifies this item as Power Management capabilities.	01h	R/O

PM Next Capabilities Pointer Register (Offset 0x49)

This register points to the next item in the Capabilities List.

Table 160: PM Next Capabilities Pointer Register (Offset 0x49)

Bit	*Field*	*Description*	*Init*	*Access*
7-0	PM Next Capabilities	Points to the next capabilities block which is for Vital Product Data (VPD).	50h	R/O

Power Management Capabilities Register (Offset 0x4A)

This 16-bit register controls various modes and features of the PCI-X device.

Table 161: Power Management Capabilities Register (Offset 0x4A)

Bit	Field	Description	Init	Access
15-11	PME Support	Indicates the power states in which the device may assert $\overline{\text{PME}}$. A 0 for any bit indicates that the device is not capable of asserting the $\overline{\text{PME}}$ pin signal while in that power state. Bit 11: $\overline{\text{PME}}$ can be asserted from D0. Default is 0. Bit 12: $\overline{\text{PME}}$ can be asserted from D1. Default is 0. Bit 13: $\overline{\text{PME}}$ can be asserted from D2. Default is 0. Bit 14: $\overline{\text{PME}}$ can be asserted from D3Hot. Default is 1. Bit 15: $\overline{\text{PME}}$ can be asserted from D3Cold. Default depends on the presence of an Aux power supply. Auxiliary power is detected by the presence of power on the VAUX_PRSNT signal pin.	01000 if no Aux 11000 if Aux present	R/O
10	D2 Support	Indicates whether the device supports the D2 power management state. This device does not support D2, so this bit is hardwired to 0.	0	R/O
9	D1 Support	Indicates whether the device supports the D1 power management state. This device does not support D1, so this bit is hardwired to 0.	0	R/O
8-6	Aux Current	The device supports the Data Register for reporting Aux Current requirements so this field is not applicable.	000	R/O
5	DSI	Indicates that the device requires device specific initialization (beyond the PCI configuration header) before the generic class device driver is able to use it. This device hardwires this bit to 0 to indicate that DSI is not necessary.	0	R/O
4	Reserved		0	R/O
3	PME Clock	Indicates that the device relies on the presence of the PCI clock for $\overline{\text{PME}}$ operation. The device does not require the PCI clock to generate $\overline{\text{PME}}$, therefore, this bit is hardwired to 0.	0	R/O
2-0	Version	A value of 010b indicates that this function complies with revision 1.1 of the PCI Power Management Interface Spec. The device hardwires this value to 010.	010	R/O

Power Management Control/Status Register (Offset 0x4C)

This 16-bit register is used to manage the PCI device's power management state as well as to enable and monitor PME events.

Table 162: Power Management Control/Status Register (Offset 0x4C)

Bit	Field	Description	Init	Access
15	PME Status	This bit is set when the device would normally assert the $\overline{\mathrm{PME}}$ signal independent of the state of the PME Enable bit. Writing a 1 to this bit will clear it and cause the device to stop asserting $\overline{\mathrm{PME}}$ (if enabled).	0	R/W2C
14-13	Data Scale	Indicates the scaling factor that is used when interpreting the value of the Data register (offset 7 in PM capability space). The device hardwires this value to 1 to indicate a scale of 1x.	1	R/O
12-9	Data Select	Indicates which data is to be reported via the Data register (offset 7 in PM capability space).	0h	R/W
8	PME Enable	Enables the device to generate $\overline{\mathrm{PME}}$ when this bit is set to 1. When 0, $\overline{\mathrm{PME}}$ generation is disabled.	1 in BCM5703 and later devices if VAUX is present, and 0 all other cases	R/W
7-2	Reserved		00h	R/O
1-0	Power State	Indicates the current power state of the device when read. When written, it sets the device into the specified power state. • 00: D0 • 01: D1 • 02: D2 • 03: D3 If software attempts to write an unsupported, optional state to this field, the write operation must complete on the bus; however, the data is discarded and no state change occurs.	00	R/W

PMCSR-BSE Register (Offset 0x4E)

The PMCSR_BSE (PMCSR PCI to PCI Bridge Support Extensions) register is not implemented in the device.

Power Management Data Register (Offset 0x4F)

This 8-bit register provides a mechanism for the device to report state dependent operating data such as power consumed or heat dissipation. Typically, the data returned through this register is a static copy of the device's worst case DC characteristics data sheet. This data, when made available to system software, could be used to intelligently make decisions about power budgeting, cooling requirements, etc.

The type of data returned by this register is dependent on the Data Select field in the PMCSR. The Data Select field, could indicate that this register should return power consumption for states D0-D3 or power dissipation for states D0-D3. Depending upon which data point is requested, the device will retrieve the corresponding information from its internal Power Dissipation or Power Consumption registers. These registers are programmed by the device CPU at boot time with the correct power values for the various states. The device boot CPU will retrieve this power information from the NVRAM at boot time.

The value returned in this register is scaled by the Data Scale field in the PMCSR. The firmware returns the maximum wattage at gigabit speed.

Table 163: Power Management Data Register (Offset 0x4F)

Bit	Field	Description	Init	Access
7-0	PM Data Register	Contains the power management data indicated by the Data Select field in the PMCSR.	0	R/O

VITAL PRODUCT DATA CAPABILITIES

Devices that support Vital Product Data (VPD) Management must support a block of registers that is part of the Capabilities List in PCI Configuration Space. The VPD Register Block is located at offset 0x50. The device supports the following VPD registers:

VPD CAPABILITY ID REGISTER (OFFSET 0X50)

This 8-bit register identifies this item in the Capabilities List as a Vital Product Data (VPD) register set.

Table 164: VPD Capability ID Register (Offset 0x50)

Bit	Field	Description	Init	Access
7-0	VPD Capability ID	Identifies this item as Vital Product Data capabilities.	03h	R/O

VPD NEXT CAPABILITIES POINTER REGISTER (OFFSET 0X51)

This register points to the next item in the Capabilities List.

Table 165: VPD Next Capabilities Pointer Register (Offset 0x51)

Bit	Field	Description	Init	Access
7-0	VPD Next Capabilities	Points to the next capabilities block which is for Message Signalled Interrupts (MSI).	58h	R/O

VPD FLAG AND ADDRESS REGISTER (OFFSET 0X52)

The upper most bit (bit 15) of this register is a flag that indicates when the transfer between the VPD Data Register and the storage component is completed. The lower 15 bits (14-0) of the register contain the byte address of the VPD to be accessed.

Table 166: VPD Flag and Address Register (Offset 0x52)

Bit	Field	Description	Init	Access
15	Flag	Indicates when the transfer between the VPD Data Register and the storage component is completed. To read VPD information, a 0 is written to the flag bit when the address is written to the VPD Address Register. The device will then set the flag bit to 1, once the four bytes of data from the storage component have been transferred to the VPD Data register. To write VPD information, a 1 is written to the Flag bit. The device will clear this bit when the data is written.	X	R/W
14-0	VPD Address	Contains the byte address of the VPD to be accessed. Since the data register is four bytes in size, the address must be aligned on a 32-bit boundary.	X	R/W

VPD DATA REGISTER (OFFSET 0X54)

VPD data can be read through this register.

Table 167: VPD Data Register (Offset 0x54)

Bit	Field	Description	Init	Access
31-0	VPD Data	The least significant byte of the register corresponds to the byte of VPD at the address specified by the VPD Address register. Four bytes are always transferred between this register and the VPD storage component. The VPD storage component is the NVRAM. VPD data is stored in the NVRAM at offset 0x100h-0x1FFh. Refer to Appendix I of the PCI 2.2 specification more complete definition of VPD.	X	R/W

MESSAGE SIGNALED INTERRUPTS CAPABILITIES

Devices that support Message Signaled Interrupts (MSI) must support a block of registers that is part of the Capabilities List in PCI Configuration Space. The MSI Register Block is located at offset 0x58.

Typical use of MSI and, in particular, multiple MSIs is to allow multiple processors to receive interrupt information independently of the others. The actual use of these in conjunction with the various send and receive queues and status information is application dependent. The device supports the following MSI registers:

MSI CAPABILITY ID REGISTER (OFFSET 0X58)

This 8-bit register identifies this item in the Capabilities List as a Message Signaled Interrupt (MSI) register set.

Table 168: MSI Capability ID Register (Offset 0x58)

Bit	Field	Description	Init	Access
7-0	MSI Capability ID	Identifies this item as Message Signaled Interrupt capabilities.	05h	R/O

MSI NEXT CAPABILITIES POINTER REGISTER (OFFSET 0X59)

This register points to the next item in the Capabilities List.

Table 169: MSI Next Capabilities Pointer Register (Offset 0x59)

Bit	Field	Description	Init	Access
7-0	MSI Next Capabilities (for BCM5721, BCM5751, and BCM5752 only)	Points to the next capabilities block that is PCIe.	0xD0	R/O
	MSI Next Capabilities (other devices)	Points to the next capabilities block that is NULL because this is the last item in the capabilities list.	0	R/O

MESSAGE CONTROL REGISTER (OFFSET 0X5A)

This 16-bit register provides system software control over MSI. After reset, MSI is disabled (bit 0 is cleared) and the function requests servicing via its $\overline{\text{INTA}}$ pin. System software can enable MSI by setting bit 0 of this register to a 1. System software is permitted to modify the Message Control register's read/write bits and fields. A device driver is not permitted to modify the read/write bits and fields.

Table 170: Message Control Register (Offset 0x5A)

Bit	Field	Description	Init	Access
15-8	Reserved		0	R/O
7	64-bit Address Capable	Indicates that the device is capable of generating a 64-bit message address. This bit is hardwired to 1 because the device is 64-bit message address capable.	1	R/O
6-4	Multiple Message Enable	System software writes to this field to indicate the number of allocated messages (equal to or less than the number of requested message). The number of allocated messages is aligned to a power of two. When MSI is enabled, a device will be allocated at least one message. The encoding is as follows: Encoding # messages allocated 000 1 001 2 010 4 011 8 100 16 101 32 110 Rsvd 111 Rsvd	000	R/W
3-1	Multiple Message Capable	System software reads this field to determine the number of requested messages. The number of requested messages must be aligned to a power of two. The encoding is as follows: Encoding # messages allocated 000 1 001 2 010 4 011 8 100 16 101 32 110 Rsvd 111 Rsvd	011	R/O
0	MSI Enable	If 1, the function is permitted to use MSI to request service and is prohibited from using the $\overline{\text{INTA}}$ pin. If 0, the device is prohibited from using MSI. System software sets this bit to enable MSI. A device driver is prohibited from writing to this bit.	0	R/W

Message Address Register (Offset 0x5C)

This 64-bit register contains the system-specified message address. If the Message Enable bit (bit 0 of the Message Control Register) is set, the contents of this register specify a 32-bit aligned address for the MSI write transaction.

Table 171: Message Address Register (Offset 0x5C)

Bit	Field	Description	Init	Access
63-0	MSI Address Register	Contains the system-specified message address.	X	R/W

Message Data Register (Offset 0x64)

This 16-bit registers contains a system-specified message. Each MSI function is allocated up to 32 unique messages. System architecture specifies the number of unique messages supported by the system.

If the Message Enable bit (bit 0 in the Message Control register) is set, the message data is driven onto the lower word of the memory write transaction's data phase. The upper 16 bits are driven to zero during the data phase.

The Multiple Message Enable field (bits 6-4 of the Message Control register) defines the number of low-order message data bits the function is permitted to modify to generate its system software allocated messages. For example, a Multiple Message Enable encoding of 010 indicates the function has been allocated four messages and is permitted to modify message data bits 1 and 0 in order to generate up to four unique messages. This field is read/write.

Table 172: Message Data Register (Offset 0x64)

Bit	Field	Description	Init	Access
15-0	MSI Data Register	Contains the system specified message.	X	R/W

Hardware Fix Register (Offset 0x66)

This 16-bit register enables fixes for certain hardware errata in the BCM5703 B0, BCM5704 B0, and later PCI-X devices.

Table 173: Hardware Fix Register (Offset 0x66)

Bit	Field	Description	Init	Access
15-14	Reserved	-		
13	HW Fix 4	Setting this bit to 1 fixes a problem where PCI latency timer does not follow the PCI specification.	0	R/W
12	HW Fix 3	Setting this bit to 1 fixes a problem where Master abort in PCI mode violate spec ($\overline{\text{FRAME}}$ and $\overline{\text{IRDY}}$ deassert at the same time).	0	R/W
11	HW Fix 2	Setting this bit to 1 fixes a problem where PCIX latency limit up to eight clocks for signals Split Response or Retry.	0	R/W
10	HW Fix 1	Setting this bit and bit 11 to 1 fixes a problem where Data corruption by non-Quadword write with 64-bit PCIX Memory Write Block command.	0	R/W
9-0	Reserved	-		

PCI-X Split Latency Timer Register (Offset 0x66, BCM5714C, BCM5714S, BCM5715C, and BCM5715S Only)

Table 174: PCI-X Split Latency Timer Register (Offset 0x66, for BCM5714 Only)

Bit	*Field*	*Description*	*Init*	*Access*
15-10	Reserved	-		
9-0	Max_split_latency	PCI-X split latency timer value.	0	R/W

PRIVATE PCI CONFIGURATION REGISTERS

MISCELLANEOUS HOST CONTROL REGISTER (OFFSET 0X68)

The Miscellaneous Host Control register is used to control various functions within the device normally controllable from the PCI-based host. Each bit has a separate function from any other bit in this register.

Note: See "Revision Levels" on page 5. Also see the latest errata documentation for any known errata related to ASIC revision string information.

Table 175: Miscellaneous Host Control Register (Offset 0x68)

Bit	*Field*	*Description*	*Init*	*Access*
31-16	ASIC Revision	ASIC revision string (see "Revision Levels" on page 5).	xxxxh	R/O
15	Enable TLP Minor Error Tolerance (BCM5721, BCM5751, and BCM5752 only)	Set this bit to enable TLP minor error tolerance (ATTR/TC/LOCK command).	0	R/W
	Reserved	-	0	R/O
14	Log Header Overflow (BCM5721, BCM5751, and BCM5752 only)	Set this bit to enable log header due to overflow.	0	R/W
	Reserved	-	0	R/O
13	Boundary Check (BCM5721, BCM5751, and BCM5752 only)	Set this bit to enable crossing 4 KB boundary check.	0	R/W
	Reserved	-	0	R/O
12	Byte-Enable Rule Check (BCM5721, BCM5751, and BCM5752 only)	Set this bit to enable the byte-enable rule check.	0	R/W
	Reserved	-	0	R/O
11	Interrupt Check (BCM5721, BCM5751, and BCM5752 only)	Set this bit to enable the interrupt check.	0	R/W
	Reserved	-	0	R/O
10	RCB Check (BCM5721, BCM5751, and BCM5752 only)	Set this bit to enable RCB check.	0	R/W
	Reserved	-	0	R/O
9	Enable Tagged Status Mode (Other devices)	When set, a unique eight-bit tag value will be inserted into the Status Block Status Tag (see "Status Block" on page 103).	0	R/W
	Reserved (BCM5788)	-	0	R/O
8	Mask_Interrupt_Mode	When set, the INTA_L signal is masked (de-asserted) at the chip's pin. However, the internal interrupt state (host coalescing event) will not be cleared.	0	R/W
7	Enable Indirect Access	Set bit to enable indirect addressing mode.	0	R/W

Table 175: Miscellaneous Host Control Register (Offset 0x68) (Cont.)

Bit	Field	Description	Init	Access
6	Enable Register Word Swap	Set bit to enable word swapping when accessing registers through the PCI target interface.	0	R/W
5	Enable Clock Control Register read/write capability	Set bit to enable Clock Control Register read/write capability, otherwise, the Clock Control Register is read only (see "PCI Clock Control Register (Offset 0x74)" on page 334).	0	R/W
4	Enable PCI State Register read/write capability	Set bit to enable PCI State Register read/write capability, otherwise, the PCI State Register is read only (see "PCI State Register (Offset 0x70)" on page 332).	0	R/W
3	Enable Endian Word Swap	Set bit to enable endian word swapping when accessing through PCI Target interface.	0	R/W
2	Enable Endian Byte Swap	Set bit to enable endian byte swapping when accessing through PCI Target interface.	0	R/W
1	Mask PCI Interrupt Output	Setting this bit will mask (i.e., prevent) future interrupt events from causing $\overline{\text{INTA}}$ to assert as long as this bit is set. Setting this bit will not clear or de-assert the internal interrupt state, nor will it de-assert the external interrupt state on $\overline{\text{INTA}}$. In other words, setting this bit does not disable the interrupt line because $\overline{\text{INTA}}$ will stay asserted if it was already asserted. However, if $\overline{\text{INTA}}$ is not already asserted when this bit is set, $\overline{\text{INTA}}$ will not be asserted if interrupt-causing event occurs later while this bit is still set. In that scenario, the interrupt will not be presented to $\overline{\text{INTA}}$ until this bit is cleared.	0	R/W
0	Clear Interrupt $\overline{\text{INTA}}$	Setting this bit will clear (de-assert) $\overline{\text{INTA}}$ as long as the *Mask Interrupt* bit is not set. If *the Mask Interrupt* bit is set, then writing the *Clear Interrupt* bit to a 1 will not de-assert $\overline{\text{INTA}}$, however it will clear the internal unmasked interrupt state, so if $\overline{\text{INTA}}$ is later unmasked, then the $\overline{\text{INTA}}$ will de-assert. However, if the *Mask Interrupt* bit is then set again, then $\overline{\text{INTA}}$ will be asserted again, because the internal masked state of the interrupt line cannot be cleared by writing to the *Clear Interrupt* bit. Since this writing to this bit does not unconditionally clear interrupts, it is recommended that software drivers write *to Interrupt Mailbox 0* (see "Interrupt Mailbox 0 Register (Offset 0x200)" on page 372 for host standard and flat modes and "Interrupt Mailbox 0 Register (Offset 0x5800)" on page 492 for indirect mode) in order to cleanly clear interrupts.	0	W/O

DMA READ/WRITE CONTROL REGISTER (OFFSET 0X6C)

The DMA read/write Control register is used to control various DMA and PCI master functions of the device.

Table 176: DMA Read/Write Control Register (Offset 0x6C)

Bit	Field	Description	Init	Access
31-28	Default PCI Write Command (other devices)	Use this command for PCI write transactions less than four words.	7h	R/W
	30-29: Write Control Boundary (BCM5721, BCM5751, and BCM5752 only)	This field sets the write control boundary and has the following values: • 00 = Break request on a multiple of a 64-byte boundary • 01 = Break request on a multiple of a 128-byte boundary • 1X = No constraint	10b	R/W
	28: Reserved (BCM5721, BCM5751, and BCM5752 only)	–	0	R/O
27-24	Default PCI Read Command (other devices)	Use this command for PCI read transactions less than four words.	6h	R/W
	Reserved (BCM5721, BCM5751, and BCM5752)	–	0000	R/W
23	Assert all BEs on DMA Write (BCM5700, BCM5701, BCM5714, and BCM5715 only)	During DMA write operations, drive all byte enables. This forces word alignment without having to adjust the DMA channel addresses and length.	0	R/W
	PCIX-32 DMA Write Single Disconnect Fix (BCM5703 A3, BCM5704 A3, and later)	Setting this bit to 1 fixes the hardware errata for a DMA write single cycle disconnect, followed by a split completion, followed by the continuation of DMA write, which causes data duplication.	0	R/W
	Reserved (other devices)	–	0	R/O
22	Use MemRdMult Command (BCM5700, BCM5701, BCM5714, and BCM5715 only)	Use the Memory Read Multiple command in place of Memory Read Line command for DMA reads.	0	R/W
	Keep Request On (BCM5702 and Later)	When this bit is set, the device continues asserting the PCI REQ# signal until the master latency timer expires, even if GNT# is removed. This bit is used for hardware debugging only.	0	R/W
	Reserved (other devices)	–	0	R/O

Table 176: DMA Read/Write Control Register (Offset 0x6C) (Cont.)

Bit	Field	Description	Init	Access
21-19	DMA Write Watermark	Encoded bits that prevents PCI bus activity until the DMA write FIFO reaches certain watermarks. For PCI: • 0 = 32 • 1 = 64 • 2 = 96 • 3 = 128 • 4 = 160 • 5 = 192 • 6 = 224 • 7 = 256 For PCI-X: • 0 = 64 • 1 = 128 • 2 = 256 • 3 = 384 (uses only bits 19 and 20; bit 21must be 0) • 4 = 512 (BCM5714 and BCM5715 only) For PCIe: • 0 = 32 • 1 = 64 • 2 = 96 • 3 = 128 • 4 = 160 • 5 = 192 • 6 = 224 • 7 = 256	000	R/W
18-16	DMA Read Watermark (BCM5700, BCM5701, and BCM5702 only)	Encoded bits that prevents PCI bus activity until the DMA read FIFO reaches certain watermarks. For PCI: • 0 = 32 • 1 = 64 • 2 = 96 • 3 = 128 • 4 = 160 • 5 = 192 • 6 = 224 • 7 = 256 For PCI-X: • 0 = 64 • 1 = 128 • 2 = 256 • 3 = 384 (uses only bits 16 and 17; bit 18 must be 0)	000	R/W

Table 176: DMA Read/Write Control Register (Offset 0x6C) (Cont.)

Bit	*Field*	*Description*	*Init*	*Access*
18-16 (cont.)	DMA Read Watermark (BCM5703C, BCM5703S, BCM5704C, and BCM5704S only)	Encoded bits that prevents PCI bus activity until the DMA read FIFO reaches certain watermarks. For PCI: • 0 = 32 • 1 = 64 • 2 = 96 • 3 = 128 • 4 = 160 • 5 = 192 • 6 = 224 • 7 = 256 For PCI-X: • 0 = 64 • 1 = 128 • 2 = 256 • 3 = 384 • 4 = 512 • 5 = 1024 • 6 = 1536 • 7 = 1536 ***Note:*** In the BCM5703C/BCM5703S, the DMA Read Watermark should be set to less than or equal to the Maximum Memory Read Byte Count of the PCI-X Command Register (see "PCI-X Command Register (Offset 0x42)" on page 312). For example, If Maximum Memory Read Byte Count is set to 0 (i.e., 512), the allowable DMA Read Watermarks are 0 to 4.	000	R/W
	DMA Read Watermark (BCM5705 and BCM5788 only)	For PCI: • 0 = 32 • 1 = 64 • 2 = 96 • 3 = 128 • 4 = 160 • 5 = 192 • 6 = 224 • 7 = 256	000	R/W
	DMA Read Watermark (BCM5714C, BCM5714S, BCM5715C, and BCM5715S only)	For PCI-X: • 0 = 64 • 1 = 128 • 2 = 256 • 3 = Reserved • 4 = Reserved • 5 = Reserved • 6 = Reserved • 7 = Reserved	000	R/W
	Reserved (all other devices)	–	000	R/O

Table 176: DMA Read/Write Control Register (Offset 0x6C) (Cont.)

Bit	Field	Description	Init	Access
15-14	15-14: One DMA at Once (BCM5704, BCM5714, and BCM5715 only)	For BCM5704: • 00: Off (default) • 01: Service both channels requests when current DMA request from this channel is completed • 1x: Service other channels request when non-idle termination code is detected For BCM5714 and BCM5715: • 15 (ONE_DMA_AT_ONCE_LOCAL): When set to 1, allows interleaving of DMA requests from the other function, but does not allow any other DMA request from the same function. The current DMA must finish before any other request from the same function can have access. • 14 (ONE_DMA_AT_ONCE_GLOBAL): When set to 1, allows only one DMA channel to have access to the PCI bus at a time. The current DMA must finish before any other channel can have access.	00	R/W
	Reserved (other devices)	–	0	R/O
13-11	DMA Write Address Boundary	Encoded bits which force termination of DMA Write operations at certain address byte boundaries. For PCI: • 0 = Disable • 1 = 16 • 2 = 32 • 3 = 64 • 4 = 128 • 5 = 256 • 6 = 512 • 7 = 1024 For PCI-X: • 0 = Disable • 1 = 128 • 2 = 256 • 3 = 384 (uses only bits 11 and 12)	000	R/W
	Reserved (BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	–	0	R/O

Table 176: DMA Read/Write Control Register (Offset 0x6C) (Cont.)

Bit	Field	Description	Init	Access
10-8	DMA Read Address Boundary	Encoded bits which force termination of DMA Read operations at certain address byte boundaries. For PCI: • 0 = Disable • 1 = 16 • 2 = 32 • 3 = 64 • 4 = 128 • 5 = 256 • 6 = 512 • 7 = 1024 For PCI-X: • 0 = Disable • 1 = 128 • 2 = 256 • 3 = 384 (uses only bits 8 and 9)	000	R/W
	Reserved (BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	–	000	R/O
7-0	Reserved (BCM5721, BCM5751, and BCM5752 only)	–	0	R/W
	Minimum DMA (other devices)	Minimum number of PCI words/double words that each DMA channel is allowed to keep the PCI bus without allowing accesses by the other DMA channel. This guarantees a minimum PCI usage rather than the usual alternate per burst behavior.	0	R/W

PCI State Register (Offset 0x70)

The PCI State register is used to control several functions within the device associated with the PCI interface.

Note: The Enable PCI State Register Read/Write Capability bit of the Miscellaneous Host Control register must be enabled to write the PCI State register from the PCI configuration cycle (see "Miscellaneous Host Control Register (Offset 0x68)" on page 325).

Table 177: PCI State Register (Offset 0x70)

Bit	Field	Description	Init	Access
31-24	Split Completion Message Index	This field returns the PCI-X Split Completion Message Index, and is valid if the Received Split Completion Error Message bit of the PCI-X Status register (see Table 158 on page 314) is set.	0	R/O
	Reserved (BCM5721, BCM5751, and BCM5752 only)		0	R/O
23-20	Split Completion Message Class	This field returns the Split Completion Message Class, and is valid if the Received Split Completion Error Message bit of the PCI-X Status register (see Table 158 on page 314) is set.	0	R/O
	Reserved (BCM5721, BCM5751, and BCM5752 only)		0	R/O
19-17	Reserved	-	0	R/O
16	No Snoop	Set this bit to enable the PCI-X No Snoop attribute bit.	0	R/W
	Reserved (BCM5721, BCM5751, and BCM5752 only)		0	R/O
15	Config Retry (BCM5721, BCM5751, and BCM5752 only)	When asserted, forces all config access to be retried.	1	R/W
	Reserved (other devices)		0	R/O
14	Reserved	-	0	R/O
13	Reserved (BCM5700, BCM5701, BCM5721, BCM5751, BCM5752 only)	-	0	R/O
	Retry Same DMA (other devices)	When set, prevents internal arbitration logic from switching to the other DMA engine after a retry cycle.	0	R/W
12	3.3VAux Present	This bit reads as 1 when the 3.3V auxiliary power source is present.	0	R/O
11-9	Max PCI Target Retry (BCM5700, BCM5701, BCM5721, BCM5751, and BCM5752 only)	Indicates the number of PCI clock cycles before Retry occurs, in multiple of 8. At reset, this field is set to 001.	001	R/O[1]
	Reserved (other devices)	-	000	R/O
8	Flat View	Asserted if the Base Address Register presents a 32 MB PCI Address Map Flat View, otherwise, indicates a 64 KB PCI Address Map Standard View.	0	R/O[1]

Table 177: PCI State Register (Offset 0x70) (Cont.)

Bit	*Field*	*Description*	*Init*	*Access*
7	VPD Available	This bit reads as 1 if the VPD region of the NVRAM can be accessed by the host.	0	R/O
6	PCI Expansion ROM Retry	Force PCI Retry for accesses to Expansion ROM region, if enabled.	0	R/O[1]
5	PCI Expansion ROM Desired	Enable PCI ROM Base Address Register to be visible to the PCI host.	0	R/O[1]
4	32-bit PCI bus	Asserted if on a 32-bit PCI bus, otherwise, indicates a 64-bit bus. Writes to this bit will force true 32-bit PCI operation even if PCI host indicated 64-bit operation. Will cause problems if this bit is asserted on a true 64-bit PCI bus.	Depends	R/O[1]
	Reserved (BCM5721, BCM5751, and BCM5752 only)		0	R/O
3	33 MHz/66 MHz PCI bus 66 MHz/133 MHz PCI-X bus	Asserted if conventional PCI bus is operating between 33-66 MHz, otherwise, indicates operation between 0-33 MHz. Asserted if PCI-X bus is operating in 66-133 MHz mode, otherwise, indicates operation in 50-66 MHz mode.	Depends	R/O[1]
	Reserved ((BCM5721, BCM5751, and BCM5752 only)		0	R/O
2	PCI bus Mode	Asserted if PCI bus is in PCI bus mode, otherwise, indicates conventional PCI-X bus. Writes to this bit will force true PCI bus operation even if PCI host indicated conventional PCI-X bus operation.	Depends	R/O[1]
	Reserved (BCM5721, BCM5751, and BCM5752 only)		0	R/O
1	Interrupt State	This bit reflects the internal state of the PCI $\overline{\text{INTA}}$ signal. A value of 0 indicates that $\overline{\text{INTA}}$ is internally asserted. Assertion of the PCI $\overline{\text{INTA}}$ pin on the PCI bus may be masked by setting the Mask PCI Interrupt Output bit of the "Miscellaneous Host Control Register (Offset 0x68)" on page 325.	1	R/O[1]
	Reserved (BCM5721, BCM5751, and BCM5752 only)		0	R/O
0	Force PCI Reset	Will force an immediate reset of the PCI interface. All state information will be lost. This bit is self-clearing.	0	R/O[1]
	Reserved (BCM5721, BCM5751, and BCM5752 only)		0	R/O

[1] Bit-enabled R/W through PCI configuration space.

PCI Clock Control Register (Offset 0x74)

Note: The Enable Clock Control Register Read/Write Capability bit of the Miscellaneous Host Control register must be enabled to write the PCI Clock Control register from the PCI configuration cycle (see "Miscellaneous Host Control Register (Offset 0x68)" on page 325).

The below table is applicable to BCM5700, BCM5701, BCM5702, BCM5703, and BCM5704 devices only.

Table 178: PCI Clock Control Register (Offset 0x74)

Bit	Description	Comments	Init	Access
31-20	Reserved		0	R/O
19	Power down (BCM5704 only)	Write 1 to power down the device.	0	R/W
	Reserved (for other devices)		0	R/O
18	Low Speed PLL Clock	This bit along with the Alternate Clock Control field controls the CORE and CPU clock source (see Table 179 on page 336). Set for entering a low-power WOL mode. • 0: 133/66 MHz mode. • 1: 66/33 MHz mode The PLL needs at least 27 µs to stabilize after its speed is changed. See Figure 108 on page 336.	0	R/W
17	BIST Control (BCM5704 only)	Controls BIST.	0	R/W
	Reserved (for other devices)		0	R/O
16	Enable BIST	Enable BIST in manufacturing.	0	R/W
	Asynchronous BIST Reset	Resets BIST	0	R/W
15	Disable system PLL		0	R/W
	Disable PCI PLL (BCM5704 only)		0	R/W
14	Disable PCI PLL	When set, the PCI PLL is set into low-power mode and the output clock from the PLL stops. To modify this bit, the Enable Alternate clock bit must first be set. When cleared, the PLL resumes normal operation.	0	R/W
	Disable CIOB-E PLL (BCM5704 only)	When this bit is set, the CIOB-E PLL is disabled.	0	R/W

Table 178: PCI Clock Control Register (Offset 0x74) (Cont.)

Bit	Description	Comments	Init	Access
13	Select Source Of Alternate Clock	When clear, the alternate clock source is: • CK25 = XTALIN (for BCM5701, BCM5702, BCM5703, and BCM5704) • CK25 = 125 MHz TXCLKIN clock divided by 5 (for BCM5700). When set, the alternate clock source is: • MII_TXCLK (for BCM5700, BCM5701, BCM5702, and BCM5703) • CK12 = XTALIN/2 (for BCM5704) See Figure 108 on page 336.	0	R/W
12	Select Alternate Clock	When clear, the system clock is sourced from the output of the system PLL. When set, the system clock is sourced from the alternate clock source. This bit must be set before attempting to modify the PLL disable bits of this register. See Figure 108 on page 336.	0	R/W
11	TX RISC Clock Disable	Disable the clock to TX RISC. When this bit is set, the clock to TX RISC will be stopped.	0	R/W
10	RX RISC Clock Disable	Disable the clock to RX RISC. When this bit is set, the clock to RX RISC will be stopped.	0	R/W
9	Enable Low-power Clock Mode	When set, the device enters a low-power state by disabling the core clock to most of the functional blocks. This bit should be set for Wake-On-LAN mode. Additional clocks should be disabled for the lowest power mode of operation. When set, register access to the general control registers will fail.	0	R/W
8	Reserved		0	R/O
7	M66EN	Value input from pin. If high, PCI PLL is being used.		R/O[a]
6-5	Reserved			R/O
4-0	PCI Clock Speed Detected	Frequency of the PCI clock speed detected: • 0 = 33 MHz • 2 = 50 MHz • 4 = 66 MHz • 6 = 100 MHz • 7 = 133 MHz Other values are undefined.		R/O[a]

[a] Bit-enabled R/W through PCI configuration space.

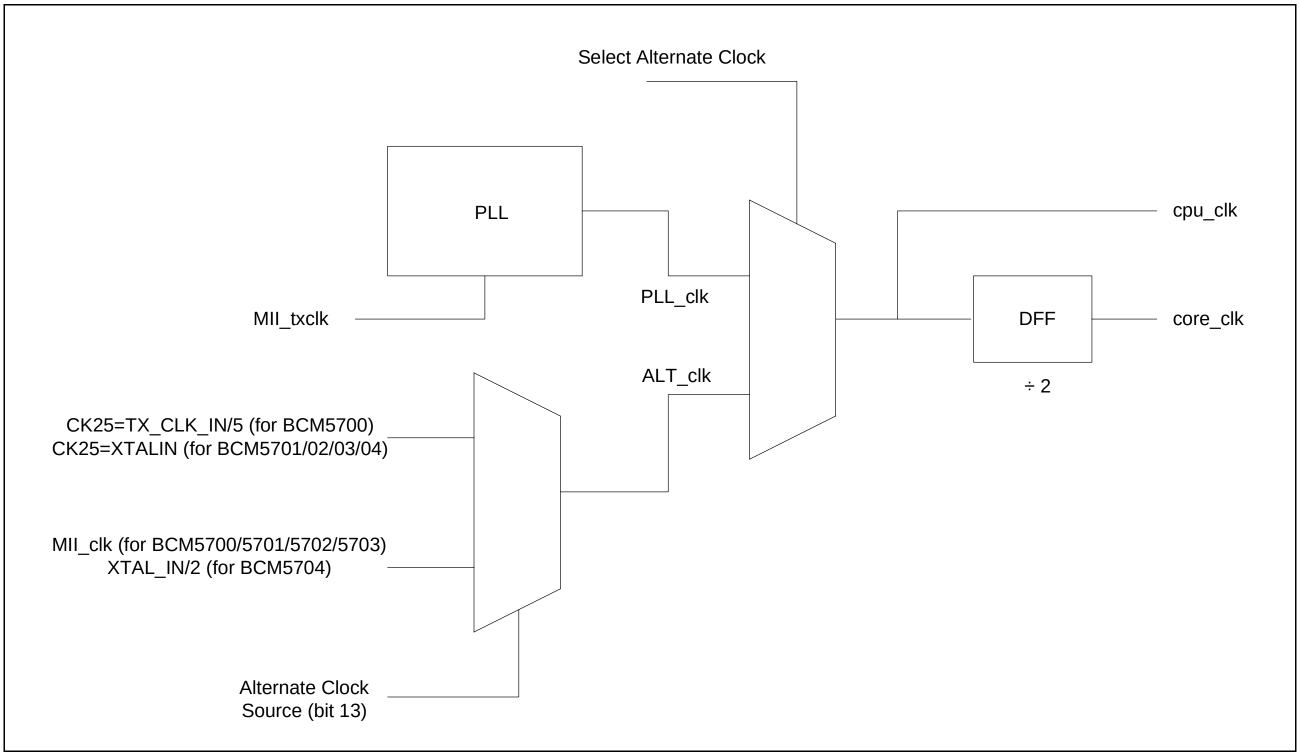


Figure 108: BCM5700/5701/5702/5703/5704 Clock Control Logic

Table 179: Clock Source and CORE_CLK Speed (BCM5704)

Bit 18	*Bit 13*	*Bit 12*	*Clock Source*	*CORE_CLK Speed (MHz)*
0	X	0	CIOBE PLL	66
1	X	0	CIOBE PLL	33
X	0	1	CK25	12.5
X	1	1	CK12	6.25

The below table is applicable to BCM5705 and BCM5788 only.

Table 180: PCI Clock Control Register Definition for BCM5705 Device

Bit	Description	Comments	Init	Access
31:27	Reserved		0	R/W
26	PCI read too long bug fix enabled	When this bit is 1, this bit enables the fix for the issue of PCI read of 8 extra bytes during the slow core clock. This bit is valid for A2 only. In A3, this bit also enables the fix for the PCI read 8 extra bytes during LSO.	0	R/W
25	PCI write too long bug fix enabled	When this bit is 1, this bit enables the fix for the issue of PCI read of 8 extra bytes during the slow core clock. This bit is valid for A2 only.	0	R/W
24	Back to A0 PCI bus arbitration when bus is parked.	When the bit is 1, the arbitration timing will be back to A0 while the bus is parked on 5705. Note that this will result in SERR# for certain chip-set, and bits 24 and 23 cannot be set simultaneously. This bit is valid for A2 only.	0	R/W
23	Bus-Parking save mode	When the bit is 1, there will always be a 2-cycle turn-around-time when a target transaction is followed by a master transaction while bus is parked on the 5705. This bit is valid for A1 only.	0	R/W
22	Reserved	This bit is changed to reserved in A3. CLKRUN# is now always enabled.	0	R/W
21	Force CLKRUN# to maintain PCI Clock	When the bit is 1, the PCI/Cardbus clock will be forced to maintain.	0	R/W
20	6.25 MHz Select	Select the 6.25 MHz clock as the alternate clock (use in Airplane Mode). If this bit is 0, the alternate clock will be selected by bit 13.	0	R/W
19	Slow Core Clock Mode	Set this bit to 1 when running a 10:1 PCI to Core clock ratio. For engineering debug only.	0	R/W
18	LED polarity	When set to 1, it would change the polarity of the 4 LEDs.	0	R/W
17	BIST Control	Controls the BIST	0	R/W
16	BIST Reset	Resets the BIST	0	R/W
15:14	Reserved		00	R/W
13	MII Clock/CK25	Use the MII CLK input as the alternate clock for the internal clocks, rather than the XTAL CK25 input as the alternate clock.	0	R/W
12	Select Alternate Clock	Use the alternate clock as the clock reference for the internal clocks, rather than the 62.5 MHz.	0	R/W
11:10	Reserved		00	R/W
9	CORE Clock Disable	Disable the CORE CLK to all blocks.	0	R/W
8	Reserved		0	R/W
7	M66EN	Value input from pin.		R/O
6	Reserved		0	R/W
5	Reserved		0	R/W
4:0	PCI Clock Speed Detected	Frequency of the PCI clock speed detected.		R/O

The below table is applicable to BCM5751, BCM5721, and BCM5752 only.

Table 181: PCI Clock Control Register Definition for BCM5751, BCM5721, and BCM5752 Devices

Bit	Description	Comments	Init	Access
31	PLP Clock Disable	When this bit is set to 1, PCIe Physical Layer Clock is disabled. N/A for PCI Device[a]	0 1 in CWOL	R/W
30	DLP Clock Disable	When this bit is set to 1, PCIe Data Link Layer Clock is disabled. N/A for PCI Device[a]	0 1 in CWOL	R/W
29	TLP Clock Disable	When this bit is set to 1, PCIe Transaction Layer Clock is disabled Once this bit is set, the FW can no longer access the PCI Config registers (including the clock control register), so it should be set only when the FW has no need to access the clock control and other PCI config registers until next power cycle. N/A for PCI Device[a]	0	R/W
28	PCIe Clock to Core Clock	When this bit is 1, the source of internal PCIe Clock is CORE_CLK. N/A for PCI Device[1]	0 1 in CWOL	R/W
27	Reserved		0	R/O
26	PCI Read Too Long ECO Fix	When this bit is 1, it enables the fix for the PCI read 8 extra bytes during LSO.	0	R/W
25	PCI Write Too Long ECO Fix	When this bit is 1, it enables the fix for PCI write 8 extra bytes during slow core clock.	0	R/W
24	Select TEST CLK Source	Selects the following test clocks: 0: CLK125 (125-MHz clock from GPHYPLL). 1: TLP CLK (/4 of the SERDES clock).	0	R/W
23	Select TEST CLK	When this bit is 1, the GPHYPLL test clock is muxed out to the GPIO0 pin.	0	R/W
22	TPM_CORE_CLK Disable	When this bit is 1, the CORE_CLK to TPM block is disabled.	0	R/W
21	Force CLKRUN	When the bit is 1, the PCI/Cardbus clock will be forced to maintain the PCI clock. N/A for PCIe Device.	0	R/W
20	Select Final Alt Clock	Select the 6.25-MHz clock as the alternate clock (use in Airplane Mode). If this bit is 0, the alternate clock will be selected by bit 13.	0	R/W
19	Slow Core Clock Mode	Set this bit to 1 when running a 10:1 PCI to Core clock ratio. For engineering debug only.	0	R/W
18	LED polarity	When set to 1, it would change the polarity of the 4 LEDs.	0	R/W
17	BIST function control	Controls the BIST function.	0	R/W
16	Asynchronous BIST Reset	Resets the BIST.	0	R/W

Table 181: PCI Clock Control Register Definition for BCM5751, BCM5721, and BCM5752 Devices (Cont.)

Bit	Description	Comments	Init	Access
15:14	Reserved			
13	Select Alt Clock Source	If set to: • 0, then alternate clock source = ck25 = (XTAL_IN)/2. • 1, then alternate clock source = ck25 = MII_CLK/2.	0	R/W
12	Select Alt Clock	Use the alternate clock as the clock reference for the internal clocks, rather than the 62.5 MHz. In BCM5752, this bit will have no effect if TPM is enabled.	0	R/W
11:10	Reserved			
9	Core Clock Disable	Disable the CORE CLK to all blocks.	0	R/W
8	Reserved			
7	M66EN	N/A for PCIe Device.		
6:5	Reserved			
4:0	Detected PCI Clock Speed	N/A for PCIe Device.		

1. Bit 28 applies to the PCIe device only. In Legacy PCI mode, PCI registers are not accessible when the device is in the D3 Cold state.

Table 182: PCI Clock Control Register Definition for BCM5714, and BCM5715 Devices

Bit	Description	Comments	Init	Access
31-27	Reserved		0	R/O
26	PCI Read Too Long ECO Fix	When this bit is 1, it enables the fix for the PCI read 8 extra bytes during LSO.	0	R/W
25	PCI Write Too Long ECO Fix	When this bit is 1, it enables the fix for PCI write 8 extra bytes during slow core clock.	0	R/W
24-21	Reserved		0	R/O
20	Select Final Alt. Clock	Select the 6.25-MHz clock as the alternate clock (use in Airplane Mode). If this bit is 0, the alternate clock will be selected by bit 13.	0	R/W
19	Slow Core Clock Mode	Set this bit to 1 when running a 10:1 PCI to Core clock ratio. For engineering debug only.	0	R/W
18	LED polarity	When set to 1, it would change the polarity of the 4 LEDs.	0	R/W
17	BIST function control	Controls the BIST function.	0	R/W
16	Asynchronous BIST Reset	Resets the BIST.	0	R/W
15:14	Reserved		0	R/O
13	Select Alt Clock Source	If set to: • 0, then alternate clock source = ck25 = (XTAL_IN)/2. • 1, then alternate clock source = ck25 = MII_CLK/2.	0	R/W
12	Select Alt Clock	Use the alternate clock as the clock reference for the internal clocks, rather than the 62.5 MHz.	0	R/W
11:10	Reserved		0	R/O
9	Core Clock Disable	Disable the CORE CLK to all blocks.	0	R/W
8	Reserved		0	R/O

Table 182: PCI Clock Control Register Definition for BCM5714, and BCM5715 Devices (Cont.)

7	M66EN	Value input from pin.	0	R/W
6:5	Reserved		0	R/O
4:0	Detected PCI Clock Speed	Frequency of the PCI clock speed detected	0	R/W

REGISTER BASE ADDRESS REGISTER (OFFSET 0X78)

The Register Base Address register defines the device local address of a register. The data pointed to by this location is read or written using the Register Data register.

To use the Register Base Address/Register Data registers:

1. Enable indirect mode by setting the Enable Indirect Access bit of the Miscellaneous Host Control register (see "Miscellaneous Host Control Register (Offset 0x68)" on page 325).
2. Write the address of the Register that you would like to access to the Register Base Address register (offset 0x78–0x7B).

   The least significant two bits of the Register Base Address register will always be ignored since Registers are naturally word (32-bit) aligned. To allow access to all of the BCM57XX registers, the range of the Register Base Address register is [17:2].
3. To write the Register pointed to by the Register Base Address register, write the 32-bit data to the Register Data register.

   At least one byte enable in the word to be written from the PCI-based Host must be asserted for the write to occur, otherwise, the write will be ignored.
4. To read the Register pointed to by the Register Base Address register, read the 32-bit data from the Register Data register.

Table 183: Register Base Address Register (Offset 0x78)

Bit	*Field*	*Description*	*Init*	*Access*
31-18	Reserved	Reserved for testing or future use.	X	R/O
17-2	Register Base Addr	Local controller address of a register than can be written or read by writing to the Register Data Register.	X	R/W
1-0	Reserved	Reserved for testing or future use.	X	R/O

Note: When using indirect register access, Broadcom recommends that the host software access the Register Base Address register (offset 0x78) and the Register Data register (offset 0x80) using PCI configuration cycles rather than memory-mapped I/O (i.e., accessing the PCI configuration registers at offsets 0x78 and 0x80 by memory addresses enabled through the PCI BAR registers). If memory-mapped I/O access is used, every register write must be followed by a read from the same register to guarantee that the posted write buffer is flushed.

MEMORY WINDOW BASE ADDRESS REGISTER (OFFSET 0X7C)

The Memory Window Base Address register defines the device local memory address which is to be the base address for the 32 KB memory window provided by the BCM57XX family. This register may contain any valid local memory address, but the usage of the least significant 15 bits varies depending on how the local memory is to be accessed. If the 32 KB memory window is used, then the least significant 15 bits are ignored. If the Memory Window Data register is referenced, then the entire Memory Window Base Address register is used to indicate the specific local memory address of the operation.

To use the Memory Window Base Address/Memory Window Data registers:

1. Enable indirect mode by setting the Enable Indirect Access bit of the Miscellaneous Host Control register (see "Miscellaneous Host Control Register (Offset 0x68)" on page 325).
2. Write the address of the Memory location that you would like to access to the Memory Window Base Address register (offset 0x7C–0x7F).

   The least significant two bits of the Memory Window Base Address register will always be ignored. Additionally, if the access to Memory is a 64-bit PCI access, the least significant three bits will be ignored, since Memory is naturally double word (64-bit) aligned. To allow access to all of the BCM57XX memory, the range of the Memory Window Base Address register is [23:2]. Also note that Target Word Swap applies to Memory accesses. This needs to be taken into account when reading or writing from the Memory Window Data register.
3. To write to the Memory location pointed to by the Memory Window Base Address register, write the 32-bit or 64-bit data to the Memory Window Data register.

   A 64-bit write to the Memory Window Data register can only occur when using a PCI Memory command and not a PCI Config command. PCI Config commands only allow for 32-bit data transfers. Also note that the particular bytes that are to be written must have their associated byte enables set. At least one byte enable in the word(s) to be written from the PCI-based Host must be asserted for the write to occur, otherwise, the write will be ignored.
4. To read the Memory location pointed to by the Memory Window Base Address register, read the 32-bit or 64-bit data from the Memory Window Data register.

   The value that is written to the Memory Window Base Address register is also used for the 32-KB Memory Window. The offset of the Memory Window into the Memory space can be determined by zeroing Memory Window Base Address register bits[14:0].

Table 184: Memory Window Base Address Register (Offset 0x7C)

Bit	Field	Description	Init	Access
31-24	Reserved	-	0	R/O
23-2	Memory Window Base Addr	Local controller memory address of the NIC memory region that can be accessed via Memory Window Data register.	X	R/W
1-0	Reserved	-	0	R/O

REGISTER DATA REGISTER (OFFSET 0X80)

The Register Data register is used to access registers in the BCM57XX family. If this register is written to, the underlying register is also written. If this register is read, the current value of the underlying register is also read.

Table 185: Register Data Register (Offset 0x80)

Bit	Field	Description	Init	Access
31-0	Register Data	Register data at the location pointed to by the Register Base Address Register.	X	R/W

MEMORY WINDOW DATA REGISTER (OFFSET 0X84)

The Memory Window Data register is normally used to access locations in the local memory when the 32 KB memory window provided by the BCM57XX family is not accessible. An example of this state is during the preboot execution environment when a 64 KB memory mapping is not available, and all accesses to the device must be done through PCI Configuration space.

This register combined with the Memory Window Base Address register (see "Memory Window Base Address Register (Offset 0x7C)" on page 341) provides an indirect method to access the entire local memory address space. If this register is written to, the underlying memory location is also written. If this register is read, the current value of the underlying memory location is also read.

Table 186: Memory Window Data Register (Offset 0x84)

Bit	Field	Description	Init	Access
31-0	Memory Window Data	Memory value at the location pointed to by the Memory Window Base Address Register.	X	R/W

Note: Programmers should take special care not to read from the Memory Window Data register if the value of the Memory Window Base Address register (see "Memory Window Base Address Register (Offset 0x7C)" on page 341) is not set to a valid local memory address. Doing so will cause the BCM57XX to hang.

MODE CONTROL REGISTER (OFFSET 0X88, HOST CPU VIEW)

This register is a mirror of 0x6800 (See "Mode Control Register (Offset 0x6800)" on page 502.

MISCELLANEOUS CONFIGURATION REGISTER (OFFSET 0X8C, HOST CPU VIEW)

This register is a mirror of 0x6804 (See "Miscellaneous Configuration Register (Offset 0x6804)" on page 504.

MISCELLANEOUS LOCAL CONTROL REGISTER (OFFSET 0X90, HOST CPU VIEW)

This register is a mirror of 0x6808 (See "Miscellaneous Local Control Register (Offset 0x6808)" on page 507.

EXPANSION ROM REGISTERS (INTERNAL RISC CPU VIEW ONLY)

EXPANSION ROM BAR SIZE REGISTER (OFFSET 0X88)

This register is not applicable to the BCM5700 or BCM5701 devices. This register is for internal CPU use only.

Table 187: Expansion ROM BAR Size Register (0x88)

Bit	Field	Description	Init	Access
31-4	Reserved	Reserved.	0	R/O
3-0	BAR Size	0000: 64 KB 0001: 128 KB 0010: 256 KB 0011: 512 KB 0100: 1 MB 0101: 2 MB 0110: 4 MB 0111: 8 MB 1000: 16 MB	0000	R/W–CPU None–PCI

EXPANSION ROM ADDRESS REGISTER (OFFSET 0X8C)

This register is not applicable to the BCM5700 or BCM5701 devices.

This Register is for internal CPU use only.

Table 188: Expansion ROM Address Register (Offset 0x8C)

Bit	Field	Description	Init	Access
31	ROM Req.	ROM Request. The PCI Interface Block will set the ROM Req whenever it detects that the Host is accessing the Serial EPROM memory space.	0	R/ W–CPU[1]
30-24	Reserved	Reserved.	0	R/O
23-0	Address	Expansion ROM Address. Address field of the Memory Read transaction is loaded into this field if the address matches the Expansion ROM address space as defined by the Expansion ROM BAR and size registers. The internal processors utilize this field to execute the ROM read.	0	R/ W–CPU[1]

[1] PCI Access via Memory Read command with the ROM BAR address.

Expansion ROM Data Register (Offset 0x90)

This register is not applicable to the BCM5700 or BCM5701 devices. This register is for internal CPU use only.

Table 189: Expansion ROM Data Register (0x90)

Bit	Field	Description	Init	Access
31-0	Data	Expansion ROM Data. Loaded by the firmware after executing the ROM read cycle.	0	R/W–CPU

VPD Config Register

VPD Interface Register (Offset 0x94)

This register is not applicable to the BCM5700 or BCM5701 devices.

Table 190: VPD Interface Register (Offset 0x94)

Bit	Field	Description	Init	Access
31-1	Reserved	Reserved.	0	R/O
0	VPD Req	VPD Request. Set when the VPD Flag/Address register is written by the host. Triggers a VPD event in the CPU event register in the GRC. Cleared by the internal CPU. Writes by the host to the VPD Flag register are not accepted while this bit is set.	0	R/W[a]

[a] Not writable by PCI configuration access.

UNDI MAILBOX REGISTERS

UNDI RECEIVE BD STANDARD PRODUCER RING PRODUCER INDEX MAILBOX (OFFSET 0X98)

This is an alternate view of the Receive BD Standard Producer Ring Producer Index Mailbox from the mailboxes region. It is provided here to allow access from the PCI Configuration space.

Table 191: UNDI Receive BD Standard Producer Ring Producer Index Mailbox (Offset 0x98)

Bit	Field	Description	Init	Access
63-0	UNDI Rcv BD Std. Ring P_Idx	UNDI Rcv BD Std. Ring Producer Index Mailbox.	0h	R/W

UNDI RECEIVE RETURN RING CONSUMER INDEX MAILBOX (OFFSET 0XA0)

This is an alternate view of the Receive Return Ring 1 Consumer Index mailbox from the mailboxes region. It is provided here to allow access from the PCI Configuration space.

Table 192: UNDI Receive Return Ring Consumer Index Mailbox (Offset 0xA0)

Bit	Field	Description	Init	Access
63-0	UNDI Rcv return C_Idx	UNDI Rcv Return Ring Consumer Index Mailbox.	0h	R/W

UNDI SEND BD PRODUCER INDEX MAILBOX (OFFSET 0XA8)

This is an alternate view of the Send BD Ring 1 NIC Producer Index mailbox from the mailboxes region. It is provided here to allow access from the PCI Configuration space.

Table 193: UNDI Send BD Producer Index Mailbox (Offset 0xA8)

Bit	Field	Description	Init	Access
63-0	UNDI Send BD NIC P_Idx	UNDI Send BD NIC Producer Index Mailbox.	0h	R/W

DUAL-MAC CONTROL REGISTERS

DUAL-MAC CONTROL REGISTER (OFFSET 0XB8)

This register is applicable to the BCM5704C Dual-MAC Transceiver and BCM5704S Dual-MAC Transceiver SerDes only.

This register consists of three bits. Bits 1-0 are used by the firmware to enable the BAR and XBAR registers in the each of the target blocks. In addition, these bits are used to determine the PCI Function Number for each of the Target blocks in the BCM5704C and BCM5704S. Bit 2 is used to indicate the Mac ID. Bits 1-0 of this register are also used in conjunction with the MAC ID (Hardwired) to determine the PCI Function Number for each of the MAC's Target. This is shown in Table 195.

Table 194: Dual-MAC Control Register (Offset 0xB8)

Bit	Field	Description	Init	Access
31-3	Reserved		0	R/O
2	MAC ID	• 0: Ch0 MAC ID • 1: Ch1 MAC ID	Hardwired value.	R/O
1-0	Channel Ctl.	• 00: Ch.0 and Ch.1 Enabled (Default) • 01: Ch.1 enabled • 10: Ch. 0 enabled • 11: Ch 0 BAR/XBAR	00	R/W[a]

[a] Not writable by PCI configuration access.

Table 195: PCI Function Number

b2	b1	b0	MAC0 Func. No.	MAC1 Func. No.	Function No.	Characteristic
0	0	0	0	-	0	MAC0 and MAC1 Enabled
0	0	1	X	-	1	MAC0 Disabled, MAC1 Enabled
0	1	0	0	-	0	MAC1 Disabled, MAC0 Enabled
0	1	1	0	-	0	MAC0 and MAC1 Enabled but accessing to MAC1 register/memory space using MAC0's XBAR register
1	0	0	-	1	1	MAC0 and MAC1 Enabled
1	0	1	-	0	0	MAC0 Disabled, MAC1 Enabled
1	1	0	-	X	1	MAC1 Disabled, MAC0 Enabled
1	1	1	-	X	1	MAC0 and MAC1 Enabled but accessing to MAC1 register/memory space using MAC0's XBAR register

MAC Function Register (0XB8H)

This register is applicable to the BCM5714 and BCM5715 devices only. This register is accessible to only internal CPU.

Note: When modified, this register should be written to same value in both the MAC functions of the device.

Table 196: MAC Function Register (0XB8H)

Bit	Field	Description	Init	Access
2-0	Function Select	MAC Function modes: • 000 = MACA and MACB Enabled. • 001 = MACB Enabled. • 010 = MACA Enabled. • 011 = MACA Enabled with XBAR mode. • 100 = MACA and MACB Swapped. • 111 = Swapped turbo teaming mode.	0	R/W

MAC Message Exchange Output Register (Offset 0xBC)

This register is applicable to the BCM5704, BCM5714, and BCM5715 devices only.

Table 197: MAC Message Exchange Output Register (Offset 0xBC)

Bit	Field	Description	Init	Access
31-0	Msg Out	Each MAC consists of a 32-bit register that is used mainly to synchronize the MACs during power-down operation. Basically, MAC0's Message Exchange Register is R/W from CPU0 and MAC1's Message Exchange Register is R/W from CPU1. The output of the MAC0's Message Exchange Register is sent to MAC1 as a 32-bit discrete bus and vice versa. This is used mainly to enable each CPU to power down its MAC based on the state of the MAC Message Exchange Input Register from the other MAC.	-	R/W[1]

[1] Not writable by PCI configuration access.

MAC MESSAGE EXCHANGE INPUT REGISTER (OFFSET 0XC0)

This register is applicable to the BCM5704, BCM5714, and BCM5715 MAC Controllers only.

Table 198: MAC Message Exchange Input Register (Offset 0xC0)

Bit	Field	Description	Init	Access
31-0	Msg In	The content of the MAC0 Message Exchange Output Register is sent to the MAC1 as a discrete 32-bit bus input to the MAC1. Similarly, the content of the MAC1 Message Exchange Output Register is sent to the MAC0 as a discrete 32-bit bus input to the MAC0. As a result, the content of the 32-bit input bus can be readable by the CPU.	-	R/W[1]

[1] Not writable by PCI configuration access.

CARDBUS PC CARD FUNCTION REGISTER (OFFSET 0XC0)

The following register is applicable to the BCM5721, BCM5751, and BCM5752 devices only.

Table 199: CardBus PC Card Function Event Register (Offset 0xc0)

Bit	Field	Description	Init	Access
31-16	Reserved	-	0	R/O
15	INTR	This field is set to 1 whenever the Interrupt field in the CardBus PC Card Function Force Event register (offset 0xcc) is set.	0	W2C
14-5	Reserved	-	0	R/O
4	General Wakeup	This field is set to 1 whenever the General Wakeup field in the CardBus PC Card Function Present State register (offset 0xc8) changes its state from 0 to 1.	0	W2C
3-0	Reserved	-	0	R/O

CARDBUS PC CARD FUNCTION EVENT MASK REGISTER (OFFSET 0XC4)

This register is applicable to the BCM5721, BCM5751, and BCM5752 devices only.

Table 200: CardBus PC Card Function Event Mask Register (Offset 0xc4)

Bit	Field	Description	Init	Access
31-16	Reserved	-	0	R/O
15	Interrupt	When this bit is set to 0, the device cannot assert CINT# or Wakeup.	0	R/W
14	Wakeup	When this bit is set to 0, the device cannot assert Wakeup.	0	R/W
13-5	Reserved	-	0	R/O
4	General Wakeup	When this bit is set to 0, the device cannot assert Wakeup.	0	R/W
3-0	Reserved	-	0	R/O

CARDBUS PC CARD FUNCTION PRESENT STATE REGISTER (OFFSET 0XC8)

This register is applicable to the BCM5721, BCM5751, and BCM5752 devices only.

Table 201: CardBus PC Card Function Present State Register (Offset 0xc8)

Bit	Field	Description	Init	Access
31-16	Reserved	-	0	R/O
15	Interrupt	This field reflects the internal state of a function specific interrupt request.	0	R/O
14-5	Reserved	-	0	R/O
4	General Wakeup	This field reflects the current state of the Wakeup event(s) that are not represented by Interrupt, Write Protect, Ready, or Battery Voltage Detect fields.	0	R/O
3-2	Battery Voltage Detect	This field reflects the current state of the card's battery. • 11 = Battery operational. • 01 = Battery needs to be replaced. • 00, 01 = Battery is not providing operational voltage.	11	R/O
1	Ready	When this field is read as: • 1 = The function is ready to perform a new operation. • 0 = The function is busy.	1	R/O
0	Write Protect	This field reflects the current state of the Write Protect switch. When this field is read as 1, the card is write protected.	X	R/O

CARDBUS PC CARD FUNCTION FORCE EVENT STATE REGISTER (OFFSET 0XCC)

This register is applicable to the BCM5721, BCM5751, and BCM5752 devices only.

Table 202: CardBus PC Card Function Force Event State Register (Offset 0xcc)

Bit	Field	Description	Init	Access
31-0	Reserved	-	0	R/O

PCIe Capabilities

This register is applicable to BCM5721, BCM5751, and BCM5752 devices only.

PCIe devices include new status and control registers that are located in the Capabilities List in the device's PCI Configuration Space. These PCIe capabilities registers start at offset 0xD0 of PCI Configuration Space. These registers appear in Configuration Space regardless of whether the device is operating in PCIe mode.

PCIe Capability List Register (Offset 0xD0)

This eight-bit register identifies this item in the Capabilities List as a PCIe register set. This value is hardwired to 0x10 to indicate the PCIe capabilities set. This register is applicable to BCM5721, BCM5751, and BCM5752 devices only.

Table 203: PCIe Capability ID Register (Offset 0xD0)

Bit	Field	Description	Init	Access
7-0	PCIe Capability ID	Identifies this item as PCIe capabilities.	10h	R/O

PCIe Next Capabilities Pointer Register (Offset 0xD1)

This eight-bit register points to the next item in the Capabilities List. This register is applicable to BCM5721, BCM5751, and BCM5752 devices only.

Table 204: PCIe Next Capabilities Pointer Register (Offset 0xD1)

Bit	Field	Description	Init	Access
7-0	PCIe Next Capabilities	Points to the next capabilities block, which is NULL, because this is the last item in the capabilities list.	00h	R/O

PCIe Capabilities Register (Offset 0xD2)

This register is applicable to BCM5721, BCM5751, and BCM5752 devices only.

Table 205: PCIe Capabilities Register (Offset 0xD2)

Bit	Field	Description	Init	Access
15-14	Reserved	-	0	R/W
13-9	Interrupt Message Number	This register contains the MSI Data value that is written to the MSI destination address when any status bit in either the Slot Status register or the Root Status register is set.	0	R/W[1]
8	Slot Implemented	This register is hardwired to 0 because this is an endpoint device.	0	R/O
7-4	Device/Port Type	This register is hardwired to 0 because this is an endpoint device.	0	R/O
3-0	Capability Version	This register indicates the version of the PCIe Capability structure.	1	R/O

[1] Writable by internal processors.

DEVICE CAPABILITIES REGISTER (OFFSET 0XD4)

This register defines operational characteristics that are globally applicable to this device. This register is applicable to BCM5721, BCM5751, and BCM5752 devices only.

Table 206: Device Capabilities Register (Offset 0xD4)

Bit	*Field*	*Description*	*Init*	*Access*
31-28	Reserved	-	0	R/O
27-26	Captured Slot Power Limit Scale	This value specifies the scale used for the Power Limit. • 00 = 1.0x • 01 = 0.1x • 10 = 0.01x • 11 = 0.001x	0	R/O
25-18	Captured Slot Power Limit Value	This value specifies the upper limit on the power supplied for this device.	0	R/O
17-15	Reserved	-	0	R/O
14	Power Indicator Present	When set to 1, this value indicates that a Power Indicator is implemented on this device.	0	R/W[1]
13	Attention Indicator Present	When set to 1, this value indicates that an Attention Indicator is implemented for this device.	0	R/W1
12	Attention Button Present	When set to 1, this value indicates that an Attention Button is implemented for this device.	0	R/W1
11-9	Endpoint L1 Acceptable Latency	This value returns the latency that this device can accept when transitioning from the L1 to the L0 state. • 000 = Less than 1 µs • 001 = 1 µs to less than 2 µs • 010 = 2 µs to less than 4 µs • 011 = 4 µs to less than 8 µs • 100 = 8 µs to less than 16 µs • 101 = 16 µs to less than 32 µs • 110 = 32 µs to 64 µs • 111 = Greater than 64 µs	7h	R/W1
8-6	Endpoint L0s Acceptable Latency	This value returns the total latency that this device can accept when transitioning from the L0s to L0 state. • 000 = Less than 64 ns • 001 = 64 ns to less than 128 ns • 010 = 128 ns to less than 256 ns • 011 = 256 ns to less than 512 ns • 100 = 512 ns to less than 1 µs • 101 = 1 µs to less than 2 µs • 110 = 2 µs to 4 µs • 111 = Greater than 4 µs	6h	R/W1
5	Extended Tag Field Supported	This value returns the maximum supported tag field size when this function acts as a requester. • 0 = 5-bit tag field • 1 = 8-bit tag field	1	R/O
4-3	Phantom Functions Supported	This value is hardwired to 0 to indicate that this device only supports a single function.	0	R/O

Table 206: Device Capabilities Register (Offset 0xD4) (Cont.)

Bit	Field	Description	Init	Access
2-0	Max Payload Size Supported	This value returns the maximum data payload size (in bytes) that this function supports for TLPs. • 0 = 128 • 1 = 256 • 2 = 512 • 3 = 1024 • 4 = 2048 • 5 = 4096 • 6,7 = Reserved	5	R/W1

1. Writable by internal processors.

DEVICE CONTROL REGISTER (OFFSET 0XD8)

This register is applicable to BCM5721, BCM5751, and BCM5752 devices only.

Table 207: Device Control Register (Offset 0xD8)

Bit	Field	Description	Init	Access
15	Reserved	-	0	R/W
14-12	Max Read Request Size	This value controls the maximum read request size for this device when acting as the requester.	0x2	R/W
11	Enable No Snoop	When this bit is set, the memory accessed by this device will not be cached by the processor.	1 0 (Changed for A1)	R/W
10	Auxiliary Power PM Enable	When this bit is set, this device is enabled to draw auxiliary power independent of PME power.	0	R/O
9	Phantom Functions Enable	This bit is hardwired to 0 because this device does not support phantom functions.	0	R/O
8	Extended Tag Field Enable	When this bit is set, it enables this device to use an 8-bit Tag field as a requester.	0	R/W
7-5	Max Payload Size[1]	This value sets the maximum TLP data payload size (in bytes) for this device. • 0 = 128 • 1 = 256 • 2 = 512 • 3 = 1024 • 4 = 2048 • 5 = 4096 • 6,7 = Reserved	0	R/W
4	Enable Relaxed Ordering	When this bit is set, this device is permitted to set the Relaxed Ordering bit.	1 0 (Changed for A1)	R/W
3	Unsupported Request Reporting Enable	When this bit is set, Unsupported Request reporting is enabled.	0	R/W
2	Fatal Error Reporting Enabled	When this bit is set, Fatal Error reporting is enabled.	0	R/W

Table 207: Device Control Register (Offset 0xD8) (Cont.)

Bit	Field	Description	Init	Access
1	Non-fatal Error Reporting Enable	When this bit is set, Non-fatal Error reporting is enabled.	0	R/W
0	Correctable Error Reporting Enable	When this bit is set, Correctable Error reporting is enabled.	0	R/W

[1] The host software should not set this field above the MPS supported value of the device as advertised in bits 0-2 of register offset 0xD4. The BCM5721 and BCM5751 B1 (or previous chip versions) support only 128 MPS bytes. The C0 (or later chip versions) of the BCM5721 and the BCM5751 support up to 512 bytes of MPS.

DEVICE STATUS REGISTER (OFFSET 0XDA)

This register is applicable to BCM5721, BCM5751, and BCM5752 devices only.

Table 208: Device Status Register (Offset 0xDA)

Bit	Field	Description	Init	Access
15-6	Reserved	-	0	R/W
5	Transaction Pending	When this bit is set to 1, it indicates that this device has issued non-posted request packets which have not been completed.	0	R/O
4	Aux Power Detected	When this bit is set, it indicates that Aux power has been detected.		R/O
3	Unsupported Request Detected	When this bit is set to 1, it indicates that an Unsupported Request has been received.	0	W2C
2	Fatal Error Detected	When this bit is set to 1, it indicates that a Fatal Error has been detected.	0	W2C
1	Non-fatal Error Detected	When this bit is set to 1, it indicates that a Non-fatal Error has been detected.	0	W2C
0	Correctable Error Detected	When this bit is set to 1, it indicates that a Correctable Error has been detected.	0	W2C

LINK CAPABILITIES REGISTER (OFFSET 0XDC)

This register is applicable to BCM5721, BCM5751, and BCM5752 devices only.

Table 209: Link Capabilities Register (Offset 0xDC)

Bit	Field	Description	Init	Access
31-24	Port Number	This value indicates the port number associated with this link.	HWInit	R/O
23-18	Reserved	–	0	R/O
17-15	L1 Exit Latency	This value returns the L1 exit latency for this link.	6	R/W[1]
14-12	L0s Exit Latency	This value returns the L0s exit latency for this link. • 0 = Less that 64 ns • 1 = Less than 128 ns • 2 = Less than 256 ns • 3 = Less than 512 ns • 4 = Less than 1 µs • 5 = Less than 2 µs • 6 = Less than 4 µs • 7 = Greater than 4 µs	6	R/W1
11-10	Active State Power Management Support	This value returns the supported ASPM states. • 0 = Reserved • 1 = L0s supported • 2 = Reserved • 3 = L0s and L1 supported	1	R/W1
9-4	Maximum Link Width	This value returns the Maximum Link Width. Allowable values are 1, 2, 4, 8, 12, 16, and 32 only. All other values are reserved.	1	R/W1
3-0	Maximum Link Speed	This value returns the Maximum Link Speed. 1 = 2.5 Gbps. All other values reserved.	1	R/W1

[1] This register is writable by the internal CPU.

LINK CONTROL REGISTER (OFFSET 0XE0)

This register is applicable to BCM5721, BCM5751, and BCM5752 devices only.

Table 210: Link Control Register (Offset 0xE0)

Bit	Field	Description	Init	Access
15-8	Reserved	-	0	R/O
7	Extended Synch	When this bit is set, it forces extended sync which gives external devices (such as logic analyzers) additional time to achieve bit and symbol lock.	0	R/W
6	Common Clock Configuration	When this bit is set, it indicates that the link partners are using a common reference clock.	0	R/W
5-4	Reserved	-	0	R/O
3	Read Completion Boundary	This value indicates the Read Completion Boundary value (in bytes) of the upstream root port. • 0 = 64 • 1 = 128	0	R/W
2	Reserved	-	0	R/O
1-0	Active State Power Management Control	This value controls the Active State Power Management supported on this link. • 0 = Disabled • 1 = L0s Entry Enabled • 2 = L1 Entry Enabled • 3 = L0s and L1 Entry Enabled	0	R/W

LINK STATUS COMMAND REGISTER (OFFSET 0XE2)

This register is applicable to BCM5721, BCM5751, and BCM5752 devices only.

Table 211: Link Status Command Register (Offset 0xE2)

Bit	Field	Description	Init	Access
15-13	Reserved	-	0	R/O
12	Slot Clock Configuration	This value indicates that this device uses the same physical reference clock that the platform provides on the connector.		R/O
11-10	Reserved	-	0	R/O
9-4	Negotiated Link Width	This value returns the negotiated link width. The only valid values are 1, 2, 4, 8, 12, 16, and 32.	1	R/O
3-0	Link Speed	This value returns the negotiated link speed. 1 = 2.5 Gbps.	1	R/O

PCIe Enhanced Capabilities

These registers are applicable to BCM5721, BCM5751, and BCM5752 devices only.

PCIe devices may optionally support a new configuration space that provides an additional 4 KB of configuration registers per device. This enhanced configuration space is mapped into host memory through a 256 MB window (enabled through the Root Complex) that provides access to the 4 KB enhanced configuration space for each of the 64K possible PCIe devices. Refer to the PCIe specification for additional details on how to access the enhanced configuration space.

The offsets listed for the following registers indicate the offset from the beginning of the enhanced configuration space for that device.

Advanced Error Reporting Enhanced Capability Header Register (Offset 0x100)

This register is applicable to BCM5721, BCM5751, and BCM5752 devices only.

Table 212: Advanced Error Reporting Enhanced Capability Header Register (Offset 0x100)

Bit	Field	Description	Init	Access
31-20	Next Capability Offset	Pointer to the Virtual Channel Capability Structure	0x13C	R/O
19-16	Capability Version	This value indicates the version of this enhanced capability header.	1	R/O
15-0	Extended Capability ID	This value indicates the type of enhanced capability header for this block and is hard-wired to one to indicate the Advanced Error Reporting capability.	1	R/O

Uncorrectable Error Status Register (Offset 0x104)

This register is applicable to BCM5721, BCM5751, and BCM5752 devices only.

Table 213: Uncorrectable Error Status Register (Offset 0x104)

Bit	Field	Description	Init	Access
31-21	Reserved	-	0	R/O
20	Unsupported Request Error Status	This bit is set when an Unsupported Request Error occurs.	0	W2C
19	ECRC Error Status	This bit is set when an ECRC error occurs.	0	W2C
18	Malformed TLP Status	This bit is set when a Malformed TLP error occurs.	0	W2C
17	Receiver Overflow Status	This bit is set when a Receiver Overflow error occurs.	0	W2C
16	Unexpected Completion Status	This bit is set when an Unexpected Completion error occurs.	0	W2C
15	Completer Abort Status	This bit is set when a Completer Abort error occurs.	0	W2C
14	Completion Timeout Status	This bit is set when a Completion Timeout error occurs.	0	W2C

Table 213: Uncorrectable Error Status Register (Offset 0x104) (Cont.)

Bit	Field	Description	Init	Access
13	Flow Control Protocol Error Status	This bit is set when a Flow Control Protocol error occurs.	0	W2C
12	Poisoned TLP Status	This bit is set when a Poisoned TLP error occurs.	0	W2C
11-5	Reserved	-	0	R/O
4	Data Link Protocol Error Status	This bit is set when a Data Link Protocol error occurs.	0	W2C
3-1	Reserved	-	0	R/O
0	Training Error Status	This bit is set when a Training error occurs.	0	W2C

UNCORRECTABLE ERROR MASK REGISTER (OFFSET 0X108)

This register is applicable to BCM5721, BCM5751, and BCM5752 devices only.

Table 214: Uncorrectable Error Mask Register (Offset 0x108)

Bit	Field	Description	Init	Access
31-21	Reserved		0	R/O
20	Unsupported Request Error Mask	Setting this bit will mask Unsupported Request errors.	0	R/W
19	ECRC Error Mask	Setting this bit will mask ECRC errors.	0	R/W
18	Malformed TLP Mask	Setting this bit will mask Malformed TLP errors.	0	W2C
17	Receiver Overflow Mask	Setting this bit will mask Receiver Overflow errors.	0	R/W
16	Unexpected Completion Mask	Setting this bit will mask Unexpected Completion errors.	0	R/W
15	Completer Abort Mask	Setting this bit will mask Completer Abort errors.	0	R/W
14	Completion Timeout Mask	Setting this bit will mask Completion Timeout errors.	0	R/W
13	Flow Control Protocol Error Mask	Setting this bit will mask Flow Control Protocol errors.	0	R/W
12	Poisoned TLP Mask	Setting this bit will mask Poisoned TLP errors.	0	R/W
11-5	Reserved	-	0	R/O
4	Data Link Protocol Error Mask	Setting this bit will mask Data Link Protocol errors.	0	R/W
3-1	Reserved	-	0	R/O
0	Training Error Mask	Setting this bit will mask Training errors.	0	R/W

UNCORRECTABLE ERROR SEVERITY REGISTER (OFFSET 0X10C)

This register is applicable to BCM5721, BCM5751, and BCM5752 devices only.

Table 215: Uncorrectable Error Severity Register (Offset 0x10C)

Bit	Field	Description	Init	Access
31-21	Reserved	-	0	R/O
20	Unsupported Request Error Severity	This bit controls the severity when an Unsupported Request Error occurs. • 0 = Non-fatal • 1 = Fatal	0	R/W
19	ECRC Error Status	This bit controls the severity when an ECRC error occurs. • 0 = Non-fatal • 1 = Fatal	0	R/W
18	Malformed TLP Status	This bit controls the severity when a Malformed TLP error occurs. • 0 = Non-fatal • 1 = Fatal	1	W2C
17	Receiver Overflow Status	This bit controls the severity when a Receiver Overflow error occurs. • 0 = Non-fatal • 1 = Fatal	1	R/W
16	Unexpected Completion Status	This bit controls the severity when an Unexpected Completion error occurs. • 0 = Non-fatal • 1 = Fatal	0	R/W
15	Completer Abort Status	This bit controls the severity when a Completer Abort error occurs. • 0 = Non-fatal • 1 = Fatal	0	R/W
14	Completion Timeout Status	This bit controls the severity when a Completion Timeout error occurs. • 0 = Non-fatal • 1 = Fatal	0	R/W
13	Flow Control Protocol Error Status	This bit controls the severity when a Flow Control Protocol error occurs. • 0 = Non-fatal • 1 = Fatal	1	R/W
12	Poisoned TLP Status	This bit controls the severity when a Poisoned TLP error occurs. • 0 = Non-fatal • 1 = Fatal	0	R/W
11-5	Reserved	-	0	R/O
4	Data Link Protocol Error Status	This bit controls the severity when a Data Link Protocol error occurs. • 0 = Non-fatal • 1 = Fatal	1	R/W

Table 215: Uncorrectable Error Severity Register (Offset 0x10C) (Cont.)

Bit	Field	Description	Init	Access
3-1	Reserved	-	0	R/O
0	Training Error Status	This bit controls the severity when a Training error occurs. • 0 = Non-fatal • 1 = Fatal	1	R/W

CORRECTABLE ERROR STATUS REGISTER (OFFSET 0X110)

This register is applicable to BCM5721, BCM5751, and BCM5752 devices only.

Table 216: Correctable Error Status Register (Offset 0x110)

Bit	Field	Description	Init	Access
31-13	Reserved	-	0	R/O
12	Replay Timer Timeout Status	This bit is set when a Replay Timer Timeout error occurs.	0	W2C
11-9	Reserved	-	0	R/O
8	REPLAY_NUM Rollover Status	This bit is set when a REPLAY_NUM Rollover error occurs.	0	W2C
7	Bad DLLP Status	This bit is set when a Bad DLLP error occurs.	0	W2C
6	Bad TLP Status	This bit is set when a Bad TLP error occurs.	0	W2C
5-1	Reserved	-	0	R/O
0	Receiver Error Status	This bit is set when a Receiver error occurs.	0	W2C

CORRECTABLE ERROR MASK REGISTER (OFFSET 0X114)

This register is applicable to BCM5721, BCM5751, and BCM5752 devices only.

Table 217: Correctable Error Mask Register (Offset 0x114)

Bit	Field	Description	Init	Access
31-13	Reserved	-	0	R/O
12	Replay Timer Timeout Mask	Setting this bit will mask Replay Timer Timeout errors.	0	R/W
11-9	Reserved	-	0	R/O
8	REPLAY_NUM Rollover Mask	Setting this bit will mask REPLAY_NUM Rollover errors.	0	R/W
7	Bad DLLP Mask	Setting this bit will mask Bad DLLP errors.	0	R/W
6	Bad TLP Mask	Setting this bit will mask Bad TLP errors.	0	R/W
5-1	Reserved	-	0	R/O
0	Receiver Error Mask	Setting this bit will mask Receiver errors.	0	R/W

ADVANCED ERROR CAPABILITIES AND CONTROL REGISTER (OFFSET 0X118)

This register is applicable to BCM5721, BCM5751, and BCM5752 devices only.

Table 218: Advanced Error Capabilities and Control Register (Offset 0x118)

Bit	Field	Description	Init	Access
31-9	Reserved	-	0	R/O
8	ECRC Check Enable	Setting this bit will enable ECRC checking.	0	R/W
7	ECRC Check Capable	When this bit is set, it indicates that this device supports ECRC Checking.	1	R/O
6	ECRC Generation Enable	Setting this bit will enable ECRC generation.	0	R/W
5	ECRC Generation Capable	When this bit is set, it indicates that this device supports ECRC generation.	1	R/O
4-0	First Error Pointer	This value indicates the bit position within the "Uncorrectable Error Status Register (Offset 0x104)" on page 356 corresponding to the first error detected.	0	R/O

HEADER LOG REGISTER (OFFSET 0X118-0X12B)

These registers are applicable to BCM5721, BCM5751, and BCM5752 devices only. The Header Log Register stores the TLP header of the transaction that has incurred a failure.

VIRTUAL CHANNEL ENHANCED CAPABILITY HEADER (OFFSET 0X13C)

Table 219: Virtual Channel Enhanced Capability Header (Offset 0x13c)

Name	Bits	Access	Default Value	Description
PCIe Extended Capability ID	15:0	RO	0x0002	Extended Capability ID for the Virtual Channel Capability is 0002h
Capability Version	19:16	RO	0x1	
Next Capability Offset	31:20	RO	0x160	

Note: This register is applicable to BCM5721, BCM5751, and BCM5752 devices only.

PORT VC CAPABILITY REGISTER (OFFSET 0X140)

Table 220: Port VC Capability Register (Offset 0x140)

Name	*Bits*	*Access*	*Default Value*	*Description*
Extended VC Count	2:0	RO	0x0	Only default VC is supported
Low Priority Extended VC Count	6:4	RO	0x0	Only default VC is supported
Reference Clock	9:8	RO	0x0	Must be set to 0 for endpoint devices
Port Arbitration Table Entry Size	11:10	RO	0x0	Must be set to 0 for endpoint devices

Note: This register is applicable to BCM5721, BCM5751, and BCM5752 devices only.

PORT VC CAPABILITY REGISTER 2 (OFFSET 0X144)

Table 221: Port VC Capability Register 2 (Offset 0x144)

Name	*Bits*	*Access*	*Default Value*	*Description*
VC Arbitration Capability	7:0	RO	0x00	Field not valid when Low Priority Extended VC Count = 0.
VC Arbitration Table Offset	31:24	RO	0x00	Table not present.

Note: This register is applicable to BCM5721, BCM5751, and BCM5752 devices only.

PORT VC CONTROL REGISTER (OFFSET 0X148)

Table 222: Port VC Control Register (Offset 0x148)

Name	*Bits*	*Access*	*Default Value*	*Description*
Load VC Arbitration Table	0	RO	0	Not supported
VC Arbitration Select	3:1	RO	0x0	Not supported

Note: This register is applicable to BCM5721, BCM5751, and BCM5752 devices only.

PORT VC STATUS REGISTER (OFFSET 0X14A)

Table 223: Port VC Status Register (Offset 0x14a)

Name	Bits	Access	Default Value	Description
VC Arbitration Table Status	0	RO	0	Not supported

Note: This register is applicable to BCM5721, BCM5751, and BCM5752 devices only.

VC RESOURCE CAPABILITY REGISTER (OFFSET 0X14C)

Table 224: VC Resource Capability Register (Offset 0x14c)

Name	Bits	Access	Default Value	Description
Port Arbitration Capability	7:0	RO	0x00	Must be set to 0 for endpoint devices
Advanced Packet Switching	14	RO	0	VC may be used for non-AS packet traffic
Reject Snoop Transactions	15	RO	0	Must be set to 0 for endpoint devices
Maximum Time Slots	22:16	RO	0x00	Must be set to 0 for endpoint devices
Port Arbitration Table Offset	31:24	RO	0x00	Must be set to 0 for endpoint devices

Note: This register is applicable to BCM5721, BCM5751, and BCM5752 devices only.

VC RESOURCE CONTROL REGISTER (OFFSET 0X150)

Table 225: VC Resource Control Register (Offset 0x150)

Name	Bits	Access	Default Value	Description
TC/VC Map	7:0	RW	0xff	A 1 at bit n indicates that TC n is mapped to VC0 (bit 0 is read only and is hardwired to 1)
Load Port Arbitration Table	16	RO	0	Must be set to 0 for endpoint devices
Port Arbitration Select	19:17	RO	0x0	Must be set to 0 for endpoint devices
VC ID	26:24	RO	0x0	Default VC = 0
VC Enable	31	RO	1	Default VC is always enabled

Note: This register is applicable to BCM5721, BCM5751, and BCM5752 devices only.

VC RESOURCE STATUS REGISTER (OFFSET 0X156)

Table 226: VC Resource Status Register (Offset 0x156)

Name	Bits	Access	Default Value	Description
Port Arbitration Table Status	0	RO	0	Must be set to 0 for endpoint devices
VC Negotiation Pending	1	RO	0	1 = Flow Control Initialization for default VC still in progress (this bit always returns 0)

Note: This register is applicable to BCM5721, BCM5751, and BCM5752 devices only.

DEVICE SERIAL NO ENHANCED CAPABILITY HEADER REGISTER (OFFSET 0X160)

Table 227: Device Serial No Enhanced Capability Header Register (Offset 0x160)

Name	Bits	Access	Default Value	Description
Next Capability Offset	31:20	RO	0x16C	
Revision ID	19:16	RO	0x1	
PCIe Capability ID	15:0	RO	0x0003	

Note: This register is applicable to BCM5721, BCM5751, and BCM5752 devices only.

DEVICE SERIAL NO LOWER DW REGISTER (OFFSET 0X164)

Table 228: Device Serial No Lower DW Register (Offset 0x164)

Name	Bits	Access	Default Value	Description
reserved	31:24	RO	0xFE	
Lower MAC Address	23:0	RO	0xFFFFFF	MAC Address(23:0)

Note: This register is applicable to BCM5721, BCM5751, and BCM5752 devices only.

Device Serial No Upper DW Register (Offset 0x168)

Table 229: Device Serial No Upper DW Register (Offset 0x168)

Name	Bits	Access	Default Value	Description
Upper MAC Address	31:8	RO	0xFFFFFF	MAC Address(47:24)
reserved	7:0	RO	0xFF	

Note: This register is applicable to BCM5721, BCM5751, and BCM5752 devices only.

Power Budgeting Enhanced Capability Header Register (Offset 0x16C)

Table 230: Power Budgeting Enhanced Capability Header Register (Offset 0x16C)

Name	Bits	Access	Default Value	Description
Next Capability Offset	31:20	RO	0x000	
Revision ID	19:16	RO	0x1	
PCIe Capability ID	15:0	RO	0x0004	

Note: This register is applicable to BCM5721, BCM5751, and BCM5752 devices only.

Power Budgeting Data Select Register (Offset 0x170)

Table 231: Power Budgeting Data Select Register (Offset 0x170)

Name	Bits	Access	Default Value	Description
Reserved	31:8	RO	0x000000	
Data Select	31:0	RW from Internal CPU	0x00	Index Power Budgeting Data reported through the Data Register

Note: This register is applicable to BCM5721, BCM5751, and BCM5752 devices only.

POWER BUDGETING DATA REGISTER (OFFSET 0X174)

Table 232: Power Budgeting Data Register (Offset 0x174)

Name	Bits	Access	Default Value	Description
Reserved	31:21	RO		
Power Rail	20:18	RW from Internal CPU		Specifies the power rail of the operating condition 12V (000) 3.3V (001) 1.8V (010) Thermal (111)
Type	17:15	RW from Internal CPU		Specifies the type of the operating condition PME Aux (000) Auxiliary (001) Idle (010) Sustained (010) Maximum (111)
PM State	14:13	RW from Internal CPU		Specifies the power management state of operating condition: D0, D3
PM Sub State	12:10	RO	000	Specifies the sub states of the operating condition
Data Scale	9:8	RO	0x0	Specifies the scale to apply to the base power value
Base Power	7:0	RW from Internal CPU		Specifies in Watts the base power value in a given operating conditions

Note: This register is applicable to BCM5721, BCM5751, and BCM5752 devices only.

POWER BUDGETING CAPABILITY REGISTER (OFFSET 0X178)

Table 233: Power Budgeting Capability Register (Offset 0x178)

Name	Bits	Access	Default Value	Description
reserved	31:8	RO	0x000000	
LOM Configuration	7:0	RW from Internal CPU		Indicate that the power budget for the device is included within the system power budget Derived from NVRAM configuration If Configured as LOM, then write 1 to bit 5 of 0x7C04 else write 0

Note: This register is applicable to BCM5721, BCM5751, and BCM5752 devices only.

FIRMWARE POWER BUDGETING REGISTER 1 (OFFSET 0X17C)

Table 234: Firmware Power Budgeting Register 1 (Offset 0x17C)

Name	Bits	Access	Default Value	Description
Power Rail	15:13	RW from Internal CPU		Specifies the power rail of the operating condition 12V (000) 3.3V (001) 1.8V (010) Thermal (111)
Type	12:10	RW from Internal CPU		Specifies the type of the operating condition PME Aux (000) Auxiliary (001) Idle (010) Sustained (010) Maximum (111)
PM State	9:8	RW from Internal CPU		Specifies the power management state of operating condition: D0, D3
Base Power	7:0	RW from Internal CPU		Specifies in Watts the base power value in a given operating conditions

Note: This register is applicable to BCM5721, BCM5751, and BCM5752 devices only.

FIRMWARE POWER BUDGETING REGISTER 2 (OFFSET 0X17D)

Table 235: Firmware Power Budgeting Register 2 (Offset 0x17D)

Name	Bits	Access	Default Value	Description
Power Rail	15:13	RW from Internal CPU		Specifies the power rail of the operating condition 12V (000) 3.3V (001) 1.8V (010) Thermal (111)
Type	12:10	RW from Internal CPU		Specifies the type of the operating condition PME Aux (000) Auxiliary (001) Idle (010) Sustained (010) Maximum (111)
PM State	9:8	RW from Internal CPU		Specifies the power management state of operating condition: D0, D3
Base Power	7:0	RW from Internal CPU		Specifies in Watts the base power value in a given operating conditions

Note: This register is applicable to BCM5721, BCM5751, and BCM5752 devices only.

Firmware Power Budgeting Register 3 (Offset 0x180)

Table 236: Firmware Power Budgeting Register 3 (Offset 0x180)

Name	Bits	Access	Default Value	Description
Power Rail	15:13	RW from Internal CPU		Specifies the power rail of the operating condition 12V (000) 3.3V (001) 1.8V (010) Thermal (111)
Type	12:10	RW from Internal CPU		Specifies the type of the operating condition PME Aux (000) Auxiliary (001) Idle (010) Sustained (010) Maximum (111)
PM State	9:8	RW from Internal CPU		Specifies the power management state of operating condition: D0, D3
Base Power	7:0	RW from Internal CPU		Specifies in Watts the base power value in a given operating conditions

Note: This register is applicable to BCM5721, BCM5751, and BCM5752 devices only.

Firmware Power Budgeting Register 4 (Offset 0x182)

Table 237: Firmware Power Budgeting Register 4 (Offset 0x182)

Name	Bits	Access	Default Value	Description
Power Rail	15:13	RW from Internal CPU		Specifies the power rail of the operating condition 12V (000) 3.3V (001) 1.8V (010) Thermal (111)
Type	12:10	RW from Internal CPU		Specifies the type of the operating condition PME Aux (000) Auxiliary (001) Idle (010) Sustained (010) Maximum (111)
PM State	9:8	RW from Internal CPU		Specifies the power management state of operating condition: D0, D3
Base Power	7:0	RW from Internal CPU		Specifies in Watts the base power value in a given operating conditions

Note: This register is applicable to BCM5721, BCM5751, and BCM5752 devices only.

Firmware Power Budgeting Register 5 (Offset 0x184)

Table 238: Firmware Power Budgeting Register 5 (Offset 0x184)

Name	Bits	Access	Default Value	Description
Power Rail	15:13	RW from Internal CPU		Specifies the power rail of the operating condition 12V (000) 3.3V (001) 1.8V (010) Thermal (111)
Type	12:10	RW from Internal CPU		Specifies the type of the operating condition PME Aux (000) Auxiliary (001) Idle (010) Sustained (010) Maximum (111)
PM State	9:8	RW from Internal CPU		Specifies the power management state of operating condition: D0, D3
Base Power	7:0	RW from Internal CPU		Specifies in Watts the base power value in a given operating conditions

Note: This register is applicable to BCM5721, BCM5751, and BCM5752 devices only.

Firmware Power Budgeting Register 6 (Offset 0x186)

Table 239: Firmware Power Budgeting Register 6 (Offset 0x186)

Name	Bits	Access	Default Value	Description
Power Rail	15:13	RW from Internal CPU		Specifies the power rail of the operating condition 12V (000) 3.3V (001) 1.8V (010) Thermal (111)
Type	12:10	RW from Internal CPU		Specifies the type of the operating condition PME Aux (000) Auxiliary (001) Idle (010) Sustained (010) Maximum (111)
PM State	9:8	RW from Internal CPU		Specifies the power management state of operating condition: D0, D3
Base Power	7:0	RW from Internal CPU		Specifies in Watts the base power value in a given operating conditions

Note: This register is applicable to BCM5721, BCM5751, and BCM5752 devices only.

Firmware Power Budgeting Register 7 (Offset 0x188)

Table 240: Firmware Power Budgeting Register 7 (Offset 0x188)

Name	Bits	Access	Default Value	Description
Power Rail	15:13	RW from Internal CPU		Specifies the power rail of the operating condition 12V (000) 3.3V (001) 1.8V (010) Thermal (111)
Type	12:10	RW from Internal CPU		Specifies the type of the operating condition PME Aux (000) Auxiliary (001) Idle (010) Sustained (010) Maximum (111)
PM State	9:8	RW from Internal CPU		Specifies the power management state of operating condition: D0, D3
Base Power	7:0	RW from Internal CPU		Specifies in Watts the base power value in a given operating conditions

Note: This register is applicable to BCM5721, BCM5751, and BCM5752 devices only.

Firmware Power Budgeting Register 8 (Offset 0x18A)

Table 241: Firmware Power Budgeting Register 8 (Offset 0x18A)

Name	Bits	Access	Default Value	Description
Power Rail	15:13	RW from Internal CPU		Specifies the power rail of the operating condition 12V (000) 3.3V (001) 1.8V (010) Thermal (111)
Type	12:10	RW from Internal CPU		Specifies the type of the operating condition PME Aux (000) Auxiliary (001) Idle (010) Sustained (010) Maximum (111)
PM State	9:8	RW from Internal CPU		Specifies the power management state of operating condition: D0, D3
Base Power	7:0	RW from Internal CPU		Specifies in Watts the base power value in a given operating conditions

Note: This register is applicable to BCM5721, BCM5751, and BCM5752 devices only.

Reset Count Register (Offset 0x158)

This debug register is only applicable to BCM5752.

Table 242: Reset Count Register (Offset 0x158)

Bit	Field	Description	Init	Access
31-24	Link Down Reset	Counts link down reset events.	N/A	R/W\
23-16	PHY Hot Reset	Counts PHY hot reset events	N/A	R/W
15-8	GRC Reset	Counts GRC reset events	N/A	R/W
7-0	POR Reset	Counts POR reset events	N/A	R/W

High-Priority Mailboxes

This is a 512-byte region that contains 64 registers. These mailbox registers are:

- 64 bits for the BCM5700 MAC and BCM5701 MAC Transceivers.
- 32 bits for the rest of the BCM57XX family.

These registers are called High-Priority Mailbox registers (or high-priority mailboxes). When a value is stored in the least significant 32 bits of these registers, an event (known as a high-priority mailbox event) is generated to the one of the RX RISC or TX RISC. To write 64 bits of a mailbox location, the upper 32 bits should be written to before the lower 32 bits.

- In BCM5702 and later devices, the upper 32 bits are not used. For compatibility across the BCM57XX family, access only the lower 32 bits.
- In BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 devices, only interrupt mailbox 0 is supported. These controllers support only one send ring, one receive producer ring, and one receive return ring.

Table 243: High-Priority Mailbox Structure

Offset	31	24 23	0
0x00	Status Tag		In ISR
0x04	Not used in the BCM5702 MAC Transceiver and later		

Note: The high-priority mailbox registers are for host standard and flat modes only. For the indirect register access mode, access the mailboxes via the low-priority mailboxes (see “Low-Priority Mailboxes” on page 490).

Table 244: High-Priority Mailbox Registers

Offset	Registers
0x200-0x207	Interrupt Mailbox 0
0x208-0x20f	Interrupt Mailbox 1
0x210-0x217	Interrupt Mailbox 2
0x218-0x21f	Interrupt Mailbox 3
0x220-0x227	General Mailbox 1
0x228-0x22f	General Mailbox 2

Table 244: High-Priority Mailbox Registers (Cont.)

Offset	*Registers*
0x230-0x237	General Mailbox 3
0x238-0x23f	General Mailbox 4
0x240-0x247	General Mailbox 5
0x248-0x24f	General Mailbox 6
0x250-0x257	General Mailbox 7
0x258-0x25f	General Mailbox 8
0x260-0x267	Reserved
0x268-0x26f	Receive BD Standard Producer Ring Producer Index
0x270-0x277	Receive BD Jumbo Producer Ring Producer Index
0x278-0x27f	Receive BD Mini Producer Ring Producer Index
0x280-0x287	Receive BD Return Ring 1 Consumer Index
0x288-0x28f	Receive BD Return Ring 2 Consumer Index
0x290-0x297	Receive BD Return Ring 3 Consumer Index
0x298-0x29f	Receive BD Return Ring 4 Consumer Index
0x2a0-0x2a7	Receive BD Return Ring 5 Consumer Index
0x2a8-0x2af	Receive BD Return Ring 6 Consumer Index
0x2b0-0x2b7	Receive BD Return Ring 7 Consumer Index
0x2b8-0x2bf	Receive BD Return Ring 8 Consumer Index
0x2c0-0x2c7	Receive BD Return Ring 9 Consumer Index
0x2c8-0x2cf	Receive BD Return Ring 10 Consumer Index
0x2d0-0x2d7	Receive BD Return Ring 11 Consumer Index
0x2d8-0x2df	Receive BD Return Ring 12 Consumer Index
0x2e0-0x2e7	Receive BD Return Ring 13 Consumer Index
0x2e8-0x2ef	Receive BD Return Ring 14 Consumer Index
0x2f0-0x2f7	Receive BD Return Ring 15 Consumer Index
0x2f8-0x2ff	Receive BD Return Ring 16 Consumer Index
0x300-0x307	Send BD Ring 1 Host Producer Index
0x308-0x30f	Send BD Ring 2 Host Producer Index
0x310-0x317	Send BD Ring 3 Host Producer Index
0x318-0x31f	Send BD Ring 4 Host Producer Index
0x320-0x327	Send BD Ring 5 Host Producer Index
0x328-0x32f	Send BD Ring 6 Host Producer Index
0x330-0x337	Send BD Ring 7 Host Producer Index
0x338-0x33f	Send BD Ring 8 Host Producer Index
0x340-0x347	Send BD Ring 9 Host Producer Index
0x348-0x34f	Send BD Ring 10 Host Producer Index
0x350-0x357	Send BD Ring 11 Host Producer Index
0x358-0x35f	Send BD Ring 12 Host Producer Index
0x360-0x367	Send BD Ring 13 Host Producer Index

Table 244: High-Priority Mailbox Registers (Cont.)

Offset	***Registers***
0x368-0x36f	Send BD Ring 14 Host Producer Index
0x370-0x377	Send BD Ring 15 Host Producer Index
0x378-0x37f	Send BD Ring 16 Host Producer Index
0x380-0x387	Send BD Ring 1 NIC Producer Index
0x388-0x38f	Send BD Ring 2 NIC Producer Index
0x390-0x397	Send BD Ring 3 NIC Producer Index
0x398-0x39f	Send BD Ring 4 NIC Producer Index
0x3a0-0x3a7	Send BD Ring 5 NIC Producer Index
0x3a8-0x3af	Send BD Ring 6 NIC Producer Index
0x3b0-0x3b7	Send BD Ring 7 NIC Producer Index
0x3b8-0x3bf	Send BD Ring 8 NIC Producer Index
0x3c0-0x3c7	Send BD Ring 9 NIC Producer Index
0x3c8-0x3cf	Send BD Ring 10 NIC Producer Index
0x3d0-0x3d7	Send BD Ring 11 NIC Producer Index
0x3d8-0x3df	Send BD Ring 12 NIC Producer Index
0x3e0-0x3e7	Send BD Ring 13 NIC Producer Index
0x3e8-0x3ef	Send BD Ring 14 NIC Producer Index
0x3f0-0x3f7	Send BD Ring 15 NIC Producer Index
0x3f8-0x3ff	Send BD Ring 16 NIC Producer Index

INTERRUPT MAILBOX 0 REGISTER (OFFSET 0X200)

This mailbox register provides two functions:

- Whenever the host writes to this register, the Interrupt State is cleared, regardless of what value is written to this register. This applies to both the internal interrupt state, and the maskable external interrupt state ($\overline{\text{INTA}}$). For instance, if an interrupt-causing event had previously occurred, but interrupts were masked (i.e., the *Mask Interrupt* bit in the *Miscellaneous Host Control Register* was set when the event occurred), an interrupt would be pending internally. However, writing any value to Interrupt Mailbox 0, would clear that internally pending interrupt. Thus, when interrupts were later unmasked, $\overline{\text{INTA}}$ would not be asserted due to that event, because the event would have been cleared by the write to this register.
- Whenever In_ISR bits in this register contain a nonzero value, it indicates to the BCM57XX family that host software is in its interrupt processing routine (ISR). This causes the device to use the during interrupt coalescing registers as opposed to the non-during interrupt coalescing registers. In addition, since the device thinks the host is running its ISR, the device will not assert an interrupt if a status block is written back while this register contains a nonzero value. This provides host software with the flexibility of another mechanism to reduce interrupts (see "Host Coalescing Control Registers" on page 450).

Since interrupts are prevented when this register is a nonzero value, and since interrupts are cleared whenever this register is written (even if it is written to 0), care must be taken by the host driver to ensure that events that normally cause interrupts are not lost. In other words, if this register is set to a nonzero value during the ISR, and then set to 0 near the end of the ISR, host software should ensure that any events that occurred while in the ISR are noted. The host could do this by checking the status block again at the bottom of the ISR and scheduling another interrupt processing routine if the status block was updated with events that had not been previously handled by the host driver (see "Interrupt Processing (Not Applicable to BCM5700)" on page 296).

OTHER INTERRUPT MAILBOX REGISTERS (OFFSET 0X208-0X218)

These registers are not applicable to the BCM5705, BCM5788, BCM5721, BCM5751, BCM5751, BCM5714, and BCM5715 devices.

These registers provide one of the two pieces of functionality that *Interrupt Mailbox 0* provides. Specifically, if these registers are set to a nonzero value, they indicate to the device that host is in its ISR. This has the same effect as when *Interrupt mailbox 0* is set to a nonzero value. However, writing any value to mailboxes 1-7 does *not* clear interrupts.

If a mailbox is zero, however, this indicates the host is not in the interrupt handler. The Host Coalescing engine uses this information to determine which set of coalescing parameters it should use (see "Host Coalescing Control Registers" on page 450).

GENERAL MAILBOX REGISTERS 1-8 (OFFSET 0X220-0X258)

These registers are not applicable to the BCM5705, BCM5788, BCM5721, BCM5751, BCM5751, BCM5714, and BCM5715 devices. These are general-purpose mailboxes that are available for use by the RISCs.

RECEIVE BD STANDARD PRODUCER RING INDEX REGISTER (OFFSET 0X268)

The Receive BD Standard Producer Ring Index register contains the index of the next buffer descriptor for the standard producer ring that will be produced in the host for the NIC to DMA into NIC memory. Host software writes this register whenever it updates the standard producer ring. This register must be initialized to 0.

RECEIVE BD JUMBO PRODUCER RING INDEX REGISTER (OFFSET 0X270)

These registers are not applicable to the BCM5705, BCM5788, BCM5721, BCM5751, BCM5751, BCM5714, and BCM5715 devices.

The Receive BD Jumbo Producer Ring Index register contains the index of the next buffer descriptor for the jumbo producer ring that will be produced in the host for the NIC to DMA into NIC memory. Host software writes this register whenever it updates the jumbo producer ring. This register must be initialized to 0.

RECEIVE BD MINI PRODUCER RING INDEX REGISTER (OFFSET 0X278)

This is only applicable to BCM5700 device.

The Receive BD Mini Producer Ring Index register contains the index of the next buffer descriptor for the mini producer ring that will be produced in the host for the NIC to DMA into NIC memory. Host software writes this register whenever it updates the mini producer ring. This register must be initialized to 0.

RECEIVE BD RETURN RING 1-16 CONSUMER INDICES REGISTERS (OFFSET 0X280-0X2F8)

The Receive BD Return Ring Index register contains the index of last the buffer descriptor for a given return ring that has been consumed. Host software writes this register whenever it updates the given return ring. This register must be initialized to 0. Because the BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 devices only support a single Receive BD Return Ring, register offsets 0x288 to 0x2f8 are reserved for these controllers.

SEND BD RING 1-16 HOST PRODUCER INDICES REGISTERS (OFFSET 0X300-0X378)

The Send BD Ring Host Producer Index register contains the index of the next buffer descriptor for a given send ring that will be produced in the host for the NIC to DMA into NIC memory. Host software writes this register whenever it updates the given send ring. This register must be initialized to 0. The host producer indices may not be used at the same time as the NIC producer indices. Because the BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 devices only support a single Send Ring, register offsets 0x308 to 0x378 are reserved for these controllers.

SEND BD RING 1-16 NIC PRODUCER INDICES REGISTERS (OFFSET 0X380-0X3F8)

The Send BD Ring NIC Producer Index register contains the index of the next buffer descriptor for a given send ring that will be produced in the host directly into NIC memory. Host software writes this register whenever it updates the given send ring. This register must be initialized to 0. The host producer indices may not be used at the same time as the NIC producer indices. Because the BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 devices only support a single Send Ring, register offsets 0x388 to 0x3F8 are reserved for these controllers.

ETHERNET MAC CONTROL REGISTERS

These registers are used to control the operation of the Ethernet MAC. There are several parameters which are available for performance and compatibility tuning.

Table 245: Ethernet MAC Control Registers

Offset	*Registers*	*Init*
0x400-0x403	Ethernet MAC Mode	0x00000008
0x404-0x407	Ethernet MAC Status	0x00000000
0x408-0x40b	Ethernet MAC Event Enable	0x00000000
0x40c-0x40f	LED Control	0x82000000
0x410-0x413	MAC Address High 1	0x00000000
0x414-0x417	MAC Address Low 1	0x00000000
0x418-0x41b	MAC Address High 2	0x00000000
0x41c-0x41f	MAC Address Low 2	0x00000000
0x420-0x423	MAC Address High 3	0x00000000
0x424-0x427	MAC Address Low 3	0x00000000
0x428-0x42b	MAC Address High 4	0x00000000
0x42c-0x42f	MAC Address Low 4	0x00000000
0x430-0x433	WOL pattern Pointer	0x00000000
0x434-0x437	WOL Pattern Configuration	0x00000000
0x438-0x43b	Transmit Random Backoff	0x00000000
0x43c-0x43f	Receive MTU Size Register	0x000005F2
0x440-0x443	Gigabit PCS Test	0x00000000
0x444-0x447	Transmit Auto-negotiation	0x00000000
0x448-0x44b	Receive Auto-negotiation	0x00000000
0x44c-0x44f	MI Communication	0x10000000
0x450-0x453	MI Status	0x00000000
0x454-0x457	MI Mode	0x000c0000
0x458-0x45b	Auto-poll Status	0x00000000
0x45c-0x45f	Transmit Mode	0x00000000
0x460-463	Transmit Status	0x00000000
0x464-0x467	Transmit Lengths	0x00000000
0x468-0x46b	Receive Mode	0x00000000
0x46c-0x46f	Receive Status	0x00000000
0x470-0x473	MAC Hash Register 0	0x00000000
0x474-0x477	MAC Hash Register 1	0x00000000
0x478-0x47b	MAC Hash Register 2	0x00000000
0x47c-0x47f	MAC Hash Register 3	0x00000000
0x480-04x483	Recv BD Rules Control Register 0	0x00000000

Table 245: Ethernet MAC Control Registers (Cont.)

Offset	Registers	Init
0x484-0x487	Recv BD Rules Mask/Value Register 0	0x00000000
0x488-0x48b	Recv BD Rules Control Register 1	0x00000000
0x48c-0x48f	Recv BD Rules Mask/Value Register 1	0x00000000
0x490-0x493	Recv BD Rules Control Register 2	0x00000000
0x494-0x497	Recv BD Rules Mask/Value Register 2	0x00000000
0x498-0x49b	Recv BD Rules Control Register 3	0x00000000
0x49c-0x49f	Recv BD Rules Mask/Value Register 3	0x00000000
0x4a0-9x4a3	Recv BD Rules Control Register 4	0x00000000
0x4a4-0x4a7	Recv BD Rules Mask/Value Register 4	0x00000000
0x4a8-0x4ab	Recv BD Rules Control Register 5	0x00000000
0x4ac-0x4af	Recv BD Rules Mask/Value Register 5	0x00000000
0x4b0-0x4b3	Recv BD Rules Control Register 6	0x00000000
0x4b4-0x4b7	Recv BD Rules Mask/Value Register 6	0x00000000
0x4b8-0x4bb	Recv BD Rules Control Register 7	0x00000000
0x4bc-0x4bf	Recv BD Rules Mask/Value Register 7	0x00000000
0x4c0-0x4c3	Recv BD Rules Control Register 8	0x00000000
0x4c4-0x4c7	Recv BD Rules Mask/Value Register 8	0x00000000
0x4c8-0x4cb	Recv BD Rules Control Register 9	0x00000000
0x4cc-0x4cf	Recv BD Rules Mask/Value Register 9	0x00000000
0x4d0-0x4d3	Recv BD Rules Control Register 10	0x00000000
0x4d4-0x4d7	Recv BD Rules Mask/Value Register 10	0x00000000
0x4d8-0x4db	Recv BD Rules Control Register 11	0x00000000
0x4dc-0x4df	Recv BD Rules Mask/Value Register 11	0x00000000
0x4e0-0x4e3	Recv BD Rules Control Register 12	0x00000000
0x4e4-0x4e7	Recv BD Rules Mask/Value Register 12	0x00000000
0x4e8-0x4eb	Recv BD Rules Control Register 13	0x00000000
0x4ec-0x4ef	Recv BD Rules Mask/Value Register 13	0x00000000
0x4f0-0x4f3	Recv BD Rules Control Register 14	0x00000000
0x4f4-0x4f7	Recv BD Rules Mask/Value Register 14	0x00000000
0x4f8-0x4fb	Recv BD Rules Control Register 15	0x00000000
0x4fc-0x4ff	Recv BD Rules Mask/Value Register 15	0x00000000
0x500-0x503	Receive Rules Configuration Register	0x00000000
0x504-0x507	Low Watermark Maximum Receive Frames Register	0x00000000
0x508-0x51f	Reserved	X
0x520-0x52f	Mac Hash Table Register	0
0x530-0x58f	Ethernet Mac Perfect Address Register	0
0x590-0x593	SerDes Control Register	0
0x594-0x597	SerDes Status Register	0
0x598-0x5ff	Reserved	X

Table 245: Ethernet MAC Control Registers (Cont.)

Offset	Registers	Init
0x600-0x60f	Reserved	X
0x610-0x623	Reserved	X
0x624-0x7ff	Reserved	X
0x800-0x867	RX Statistics Memory	0x00000000
0x868-0x87f	Reserved	X
0x880-0x8ef	TX Statistics Memory	0x00000000
0x8f0-0xbff	Reserved	0x00000000

Ethernet MAC Mode Register (Offset 0x400)

Table 246: Ethernet MAC Mode Register (Offset 0x400)

Bit	Field	Description	Init	Access
31-27	Reserved	-	0	R/O
26-25	26: Free Running ACPI (BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	When this bit is set, the ACPI state machine will continue running when a match is found. When this bit is clear, the ACPI state machine will halt when a match is found.	0	R/W
	25: Halt Interesting Packet PME (BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	When this bit is set, the WOL signal will not be asserted on an interesting packet match.	0	R/W
	26-25: Reserved (other devices)	-	0	R/O
24	Keep Frame in WOL (BCM5705, BCM5788, BCM5721, BCM5751, and BCM5752 only)	-		
	Reserved (other devices)	-	0	R/O
23	Enable FHDE	Enable the receive Frame Header DMA engine. Must be set for normal operation.	0	R/W
22	Enable RDE	Enable the Receive DMA engine. Must be set for normal operation.	0	R/W
21	Enable TDE	Enable Transmit DMA engine.	0	R/W
20	Reserved (BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	-	0	R/O
	Enable MIP	Enable Management Interface Programming for PHY.	0	R/W
19	ACPI Power-on Enable	Enable Wake on LAN filters when in power-down mode.	0	R/W
18	Magic Packet Detect Enable	Enable Magic Packet Detection.	0	R/W
17	Send Configs	Send config commands when in TBI mode. (See "Transmit 1000BASE-X Auto-Negotiation Register (Offset 0x444)" on page 387.)	0	R/W
16	Flush TX Statistics	Write transmit statistics to external memory. This bit is self-clearing.	0	R/W
15	Clear TX Statistics	Clear transmit statistics internal RAM. This bit is self-clearing.	0	R/W

Table 246: Ethernet MAC Mode Register (Offset 0x400) (Cont.)

Bit	Field	Description	Init	Access
14	Enable TX Statistics	Enable transmit statistics external updates.	0	R/W
13	Flush RX Statistics	Write receive statistics to external memory. This bit is self-clearing.	0	R/W
12	Clear RX Statistics	Clear receive statistics internal RAM. This bit is self-clearing.	0	R/W
11	Enable RX Statistics	Enable receive statistics external updates.	0	R/W
10	Reserved (BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	-	0	R/O
	Link Polarity (other devices)	When clear, the LINKRDY signal has positive polarity. When set, the LINKRDY signal has negative polarity.	0	R/W
9	Max Defer	Enable Max Deferral checking statistic.	0	R/W
8	Enable TX Bursting	Enable transmit bursting in gigabit half-duplex mode.	0	R/W
7	Tagged MAC Control	Allow the MAC to receive tagged MAC control packets.	0	R/W
6-5	Reserved	-	000	R/O
4	Loopback Mode	When set, an internal loopback path is enabled from the transmit MAC to the receive MAC. This bit is provided for diagnostic purposes only.		
3-2	Port mode	These bits determine what interface the port is running: • 11: TBI (ten bit interface) • 10: GMII • 01: MII • 00: None (default)	01[1]	R/W
1	Half-duplex	When set, the MII/GMII interface is configured to operate in half-duplex mode and the CSMA/CD state machines in the MAC are set to half-duplex mode. The default value is 0.	0	R/W
0	Global Reset	When this bit is set to 1 the MAC state machine is reset. This is a self-clearing bit. The default value is 0.	0	R/W

[1] The default value after reset for the BCM5700 MAC prior to the C0 revision (see "Revision Levels" on page 5) is 10.

ETHERNET MAC STATUS REGISTER (OFFSET 0X404)

Table 247: Ethernet MAC Status Register (Offset 0x404)

Bit	Field	Description	Init	Access
31-29	Reserved	Always 0.	0	R/O
28	Interesting Packet PME Attention(BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	When this bit is set, the WOL signal is asserted when an interesting packet is detected.	0	W2C
	Reserved	Always 0.	0	R/O
27	TX Statistic Overrun	Transmit Statistics block has overrun. Generates an attention when enabled.	0	W2C
26	RX Statistic Overrun	Receive Statistics block has overrun. Generates an attention when enabled.	0	W2C
25	ODI Error	Output Data Interface block has an overrun or underrun. Will generate attention when enabled. Clear this attention using the Transmit Status Register.	0	R/O
24	AP Error	Auto-polling interface needs service. Generates an attention when enabled. Clear this attention using the Auto-polling Status Register.	0	R/O
23	MI Interrupt	Management Interface is signalling an interrupt. Generates an attention when enabled.	0	R/O
22	MI Completion	Management Interface transaction has completed. Generates an attention when enabled.	0	W2C
21-13	Reserved	-	0	R/O
12	Link State Changed	Set when the link state has changed. Generates an attention when enabled by bit 12 of the Ethernet MAC Event Enable register (see "Ethernet MAC Event Enable Register (Offset 0x408)" on page 381). Clear this attention by writing 1 to Sync Changed (bit 4) and Config changed (bit 3).	0	R/O W2C (for BCM5705, BCM5714, BCM5715, BCM5721, BCM5751, BCM5752, and BCM5788 only)
11	Reserved	-	0	R/O
10	Reserved (BCM5705, BCM5714, BCM5721 and BCM5751 only)	-	0	R/O
	Port Decode Error (other devices)	PCS detected an encoding error. This can only occur in internal PCS mode. Generates an attention when enabled.	0	W2C
9-5	Reserved	-	0	R/O
4	Reserved (BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	-	0	R/O
	Sync Changed (other devices)	PCS sync state machine has changed state.	0	W2C

Table 247: Ethernet MAC Status Register (Offset 0x404) (Cont.)

Bit	Field	Description	Init	Access
3	Reserved (BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	-	0	R/O
	Config changed (other devices)	Gigabit port receive configuration data has changed.	0	W2C
2	Reserved (BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	-	0	R/O
	Receiving Config (other devices)	Currently receiving configuration data on gigabit port.	0	R/O
1	Reserved (BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	-	0	R/O
	Signal Detect (other devices)	In TBI mode, contains the value of the LINKRDY input pin controlled by the Link Polarity function.	0	R/O
0	Reserved (BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	-	0	R/O
	PCS synched (other devices)	In TBI mode, indicates that the internal PCS function has synchronized to the data stream. The value is qualified with Signal Detect.	0	R/O

Ethernet MAC Event Enable Register (Offset 0x408)

Table 248: Ethernet MAC Event Enable Register (Offset 0x408)

Bit	Field	Description	Init	Access
31-29	Reserved	Always 0.	0	R/O
28	Interesting Packet PME Attention Enable (BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	When this bit is set, an attention will be asserted on an interesting packet match.	0	R/W
	Reserved	Always 0.	0	R/O
27	TX Statistics Overrun	Enable attention when transmit statistics block has overrun.	0	R/W
26	RX Statistics Overrun	Enable attention when receive statistics block has overrun.	0	R/W
25	ODI Error	Enable attention when an Output Data Interface block has an overrun or underrun.	0	R/W
24	AP Error	Enable attention when the Auto-polling interface has an error.	0	R/W
23	MI Interrupt	Enable attention when the Management Interface is signaling an interrupt.	0	R/W
22	MI Completion	Enable attention when the Management Interface transaction has completed.	0	R/W
21-13	Reserved		0	R/O
12	Link State Changed	Enable attention when the link has changed state.	0	R/W
11	Reserved	-	0	R/O
10	Reserved (BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	-	0	R/O
	Port Decode Error (other devices)	Enable attention when the PCS detected an encoding error and the device is in TBI mode.	0	R/W
9-0	Reserved	Always 0.	0	R/O

LED Control Register (Offset 0x40C)

Table 249: LED Control Register (Offset 0x40C)

Bit	Field	Description	Init	Access
31	Override Blink Rate	If set, the blink rate for the Traffic LED is determined by the Blink Period field (bit 30 to bit 9). This bit is reset to 1. If not set, the blink rate assumes a Blink Period of 0x040, corresponding to approximately 15.9 Hz.	1	R/W
30-19	Blink Period	Specifies the period of each blink cycle (on + off) for Traffic LED in milliseconds. Must be a nonzero value. This 12-bit field is reset to 0x040, giving a default blink period of approximately 15.9 Hz.	000001 000000	R/W
18-16	Reserved	-	0	R/O
15	Wireless Combo Mode (BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	When this bit is set, the LED pins can be shared with a wireless device which takes over when the link is lost.	0	R/W
	Reserved (other devices)	-	0	R/O
14	Shared Traffic/Link LED Mode (BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only) (See Note 1 below)	When this bit is set, the Link LED is solid green when there is a link and blinks when there is traffic. (The LED_MODE field must be set to 00 before enabling this bit.)	1	R/W
	Reserved (other devices)	-	0	R/O
13	MAC Mode (BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only) (See Note 1 below)	When this bit is set, the traffic LED blinks only when traffic is addressed for the device. (The LED_MODE field must be set to 00 before enabling this bit.)	0	R/W
	Reserved (other devices)	-	0	R/O
12-11	LED_MODE	• 00: MAC Mode (default in BCM5700 and BCM5701)—LED signal is in active low (on) when link is established and is in high (off) when link is not established. - LINKLEDB = LED10 (from MAC core) - SPD100LEDB = LED100 (from MAC core) - SPD1000LEDB = LED1000 (from MAC core) - TRAFFICLEDB = LED_TRAFFIC (from MAC core) • 01: PHY Mode 1 (default in BCM5702 and later) —LED signal is in active low (on) when link is established and is in tristate (off) when link is not established. - LINKLEDB = Link 10 (open drain) (from PHY core) - SPD100LEDB = Link 100 (open drain) (from PHY core) - SPD1000LEDB = Link 1000(open drain) (from PHY core) - TRAFFICLEDB = PHY RCVLED or PHY XMTLED (from PHY core)	00 01	R/O

Table 249: LED Control Register (Offset 0x40C) (Cont.)

Bit	Field	Description	Init	Access
12-11 (continued)	LED_MODE	• 10: PHY Mode 2—LED signal is in active low (on) when link has a valid data or idle signal and is in high (off) when link is not established. - LINKLEDB = Link10 = Link10 (from PHY core) - SPD100LEDB = Link100 = Link100 and valid data or idle (from PHY core) - SPD1000LEDB = Link1000 = Link1000 and valid data or idle (from PHY core) - TRAFFICLEDB = RCVLED or XMTLED (from PHY core) • 11: BCM5714, BCM5715, BCM5752, BCM 5721 A1 or later, and BCM5751 A1 or later only: Same as PHY Mode 1 and is used for wireless combo mode (that is, when bit 15 of this register is set to 1). • 11: Reserved (for other devices).		
10	Traffic LED status	-	0	R/O
9	10Mbps LED status	-	0	R/O
8	100Mbps LED status	-	0	R/O
7	1000Mbps LED status	-	0	R/O
6	Traffic LED	If set along with the Override Traffic bit, the Traffic LED is turned on. If the Blink Traffic LED bit is also set, the LED will blink with blink rate specified in Override Blink Rate (bit 31) and Blink Period (bit 30-19) fields.	0	R/W
5	Blink Traffic LED	If set along with the Override Traffic bit and Traffic LED bit, the Traffic LED will blink with the blink rate specified in Override Blink rate (bit 31) and Blink Period (bit 30-19) fields.	0	R/W
4	Override Traffic LED	If set, overrides hardware control of the Traffic LED. The Traffic LED will then be controlled via bit 6 and bit 5.	0	R/W
3	10 Mbps LED	If set along with the LED Override bit, turns on the 10 Mbps LED.	0	R/W
2	100 Mbps LED	If set along with the LED Override bit, turns on the 100 Mbps LED.	0	R/W
1	1000 Mbps LED	If set along with the LED Override bit, turns on the 1000 Mbps LED.	0	R/W
0	Override Link LEDs	If set, overrides hardware control of the three link LEDs. The LEDs will then be controlled via bits 3-1.	0	R/W

Note: To enable either MAC Mode or Shared Traffic/Link LED Mode, the LED Mode (bits 12:11) must be set to MAC LED mode (0x0).

ETHERNET MAC ADDRESSES REGISTERS (OFFSET 0X410-0X42C)

The Ethernet MAC needs to be initialized with up to four 6-byte MAC addresses in order to perform hardware receive packet filtering. When operating the receive MAC in promiscuous mode, no receive filtering is performed. MAC address one is used as the source address for sending flow-control packets. The addresses are not synchronized, so they must not be set after initialization unless the MAC block is reset.

Note: The BCM5702, BCM5703C, BCM5703S, BCM5704C, and BCM5704S support 12 additional Ethernet MAC address registers (see "Ethernet MAC Perfect Address Registers (Offset 0x530-0x58F)" on page 395).

Table 250: Ethernet MAC Address High Register (Offset 0x410)

Address Offset MAC Address 0: 0x410

Address Offset MAC Address 1: 0x418

Address Offset MAC Address 2: 0x420

Address Offset MAC Address 3: 0x428

Bit	*Field*	*Description*	*Init*	*Access*
31-16	Reserved	Always 0.	0h	R/O
15-0	MAC Address High	Upper 2-bytes of this node's MAC address.	0h	R/W

Table 251: Ethernet MAC Address Low Register (Offset 0x414)

Address Offset MAC Address 0: 0x414

Address Offset MAC Address 1: 0x41C

Address Offset MAC Address 2: 0x424

Address Offset MAC Address 3: 0x42C

Bit	*Field*	*Description*	*Init*	*Access*
31-0	MAC Address Low	Lower 4-byte of this node's MAC address.	0h	R/W

WOL Pattern Pointer Register (Offset 0x430)

This version of the WOL Pattern Pointer register applies to the BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 devices only.

Table 252: WOL Pattern Pointer Register (Offset 0x430)

Bit	Field	Description	Init	Access
31-9	Reserved	-		
8-0	ACPI Pointer	Specifies the offset into the 6 KB BD memory for frame comparison. (Bits 3:0 are ignored to align the memory address to a natural 128-bit boundary.)	0x000	R/W

Rest of BCM57XX Family

This version of the WOL Pattern Pointer register applies to the rest of the BCM57XX family.

Table 253: WOL Pattern Pointer Register (Offset 0x430, Rest of BCM57XX Family)

Bit	Field	Description	Init	Access
31-0	ACPI Pointer	Specifies the offset into memory for frame comparison.	0	R/W

WOL PATTERN CONFIGURATION REGISTER (OFFSET 0X434)

This version of the WOL Pattern Pointer register applies to the BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 devices only.

Table 254: WOL Pattern Configuration Register (Offset 0x434)

Bit	Field	Description	Init	Access
31-28	Reserved	-	0	R/O
27-16	ACPI offset	Offset of a frame where the pattern comparison starts.	0	R/W
15-10	Reserved	-	0	R/O
9-0	ACPI Length	Specifies the total number of 64-bit double words inside the MISC_BD memory that are valid for ACPI. For GMII, it should have a value of 2,4,6,... For MII, it should have a value of 3,6,9,....	0x000	R/W

Rest of BCM57XX Family

This version of the WOL Pattern Configuration register applies to the rest of the BCM57XX family.

Table 255: WOL Pattern Configuration Register (Offset 0x434, Rest of BCM57XX Family)

Bit	Field	Description	Init	Access
31	Reserved (BCM5700 and BCM5701 only)	-	0	R/O
	Large Burst DMA Write Enable	When set, 5701 mode is enabled, which requires a clock rate greater than 42 MHz to support interesting packet detection at 100BASE-TX speeds.	0	R/W
30-28	Reserved	-	0	R/O
27-16	ACPI offset	Offset of a frame where the patterns comparison starts.	0	R/W
15-0	ACPI Length	Specify the length of bytes for frame comparison.	0	R/W

ETHERNET TRANSMIT RANDOM BACKOFF REGISTER (OFFSET 0X438)

This register is used to initialize the random backoff interval generator. It is implemented as a 10-bit linear feedback shift register as follows:

```
random[9:0] = (random SRL 1) XOR 321 when random 9 = 1 else (random SRL 1) XOR 0
```

If the random generator is initialized to zero, then it will always remain a zero indicating that a no backoff internal is always selected. It is recommended that this field be initialized with the same value that is written to the MAC Address Low register in order to create additional randomness to the initial seed.

Table 256: Ethernet Transmit Random Backup Register (Offset 0x438)

Bit	Field	Description	Init	Access
31-10	Reserved	Always 0.	0	R/O
9-0	Random Backoff Seed	For half-duplex, initialize with any nonzero seed.	0	R/W

Receive MTU Size Register (Offset 0x43C)

This register defines the threshold above which a frame will be marked as oversize.

Table 257: Receive MTU Size Register (Offset 0x43C)

Bit	Field	Description	Init	Access
31-16	Reserved	Always 0.	0	R/O
15-0	MTU	2-byte field which is the largest size frame that will be accepted without being marked as oversize.	05F2h	R/W

Gigabit PCS Test Register (Offset 0x440)

This register is not applicable to BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 devices.

This register is used only during testing of the physical components of the gigabit Ethernet interface. When enabled, the PCS (physical coding sublayer) continuously sends the 20-bit data pattern. This register must be initialized to zero for normal operation. It is reserved in 1000BASE-T, 100 Mbps, and 10 Mbps Ethernet modes.

Table 258: Gigabit PCS Test Register (Offset 0x440)

Bit	Field	Description	Init	Access
31-21	Reserved	Always 0.	0	R/O
20	Enable PHY test mode	Send 20-bit data pattern repeatedly.	0	R/W
19-0	PHY Test Data Pattern	20-bit pattern used during PHY testing.	0	R/W

Transmit 1000BASE-X Auto-Negotiation Register (Offset 0x444)

This register is not applicable to BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 devices.

This register contains the data sent by the transmit PHY during 1000BASE-X auto-negotiation. This register is unused in 1000BASE-T, 100BASE-T, and 10BASE-T Ethernet modes.

Table 259: Transmit 1000BASE-X Auto-Negotiation Register (Offset 0x444)

Bit	Field	Description	Init	Access
31-16	Reserved	Always 0.	0	R/O
15-0	Transmit Auto-negotiation Data	2-byte field sent during auto-negotiation. Most significant byte is sent first.	0	R/W

Note: The Send Configs bit of the Ethernet MAC Mode register (see Table 246 on page 377) must be set to use this register.

RECEIVE 1000BASE-X AUTO-NEGOTIATION REGISTER (OFFSET 0X448)

This register is not applicable to BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 devices.

This register contains the data received by the receive PHY during 1000BASE-X auto-negotiation. This register is unused in 1000BASE-T, 100BASE-T, and 10BASE-T Ethernet modes.

Table 260: Receive 1000BASE-X Auto-Negotiation Register (Offset 0x448)

Bit	Field	Description	Init	Access
31-16	Reserved	Always 0.	0	R/O
15-0	Receive Auto-negotiation Data	2-byte field received during auto-negotiation. Most significant byte is received first.	0	R/O

Note: The Send Configs bit of the Ethernet MAC Mode register (see Table 246 on page 377) must be set to use this register.

MI COMMUNICATION REGISTER (OFFSET 0X44C)

This register is used to communicate with a transceiver device through the MII/GMII management interface signals MDIO and MDC. To complete a transaction, the register values are configured for the operation, and the Start bit (bit 29) is set. When the transaction completes, the Start bit will be cleared by the device. The Read Failed bit (bit 28) can be used to determine an incomplete read transaction. The Transaction Data field should be ignored when the Read Failed bit is set.

Table 261: MI Communication Register (Offset 0x44C)

Bit	Field	Description	Init	Access
31-30	Reserved	Always 0.	0	R/O
29	Start/Busy	Set this bit to start a transaction. While it is high, it indicates that the current transaction is still ongoing. If enabled, generates an attention via the EMAC Status Register MI Completion bit (bit 22).	0	R/W
28	Read failed	When set, the transceiver device did not driver the bus during the attempted read transaction. Valid after the Start/Busy bit is cleared.	0	R/O
27-26	Command	These bits specify the transaction type: 11: Undefined. 10: Read command. 01: Write command. 00: Undefined.	00b	R/W
25-21	PHY Addr	PHY Address.	0000b	R/W
20-16	Register Address	Address of the register to be read or written.	0000b	R/W
15-0	Transaction Data	When configured for a write command, the data stored at this location is written to the PHY at the specified PHY and register address. During a read command, the data returned by the PHY is stored at this location.	0	R/W

MI STATUS REGISTER (OFFSET 0X450)

This register contains status from a PHY using the management interface. It is obtained during autopolling and will not be valid if autopolling is disabled. If auto-polling is not enabled, bit 0 must be set to enable link to the MAC state machines.

Table 262: MI Status Register (Offset 0x450)

Bit	Field	Description	Init	Access
31-2	Reserved	Always 0.	0	R/O
1	Mode10 Mbps	When read, a value of 1 indicates the transceiver device is operating in 10 Mbps mode.	0	R/W
0	Link status	The bit will generate an attention if enabled. Indicates status of the link on the transceiver device. When read, a value of 1 indicates the transceiver is linked.	0	R/W

MI MODE REGISTER (OFFSET 0X454)

This register controls autopolling on the management interface. Auto control mode sets the link state in the transmit state register.

Table 263: MI Mode Register (Offset 0x454)

Bit	Field	Description	Init	Access
31-21	Reserved		0	R/O
20-16	MI Clock Count	Counter to divide CORE_CLK (i.e., 66 MHz) to generate the MI clock. The formula is: `MI Clock = CORE_CLK/2/(MI Clock Count + 1)`	0ch	R/W
15-10	Reserved	-	0	R/O
9-5	PHY Address	This field specifies the PHY address.	00001	R/W
4	Port Polling	Set to enable autopolling of the transceiver link information from the MII Management interface. If cleared, the device will obtain the link status information from the state of the LNKRDY input signal.	0	R/W
3-2	Reserved	-	0	R/O
1	Use Short Preamble	Use short preamble while polling, if set.	0	R/W
0	Reserved	-	0	R/O

AUTOPOLLING STATUS REGISTER (OFFSET 0X458)

This register contains status of autopolling the management interface.

Table 264: Autopolling Status Register (Offset 0x458)

Bit	Field	Description	Init	Access
31-1	Reserved	Always 0.	0	R/O
0	Auto-polling Error	Indicates an autopolling error occurred if set.	0	W2C

Transmit MAC Mode Register (Offset 0x45C)

This register controls the transmit Ethernet interfaces.

Table 265: Transmit MAC Mode Register (Offset 0x45C)

Bit	Field	Description	Init	Access
31-7	Reserved	Reserved for future use.	X	R/O
7	Reserved (BCM5700 and BCM5701 only)	Reserved for future use.	X	R/O
	Link aware enable	When set, transmission of packets by the MAC is enabled only when link is up (BCM5700, BCM5701 mode).	0	R/W
6	Enable Long Pause	When set, the PAUSE time value set in the transmitted PAUSE frames is 0xFFFF. The default value for PAUSE time is 0x1FFF.	0	R/W
5	Enable Big Backoff	MAC will use larger than normal back-off algorithm.	0	R/W
4	Enable Flow Control	MAC will send 802.3x flow control frames.	0	R/W
3-2	Reserved	-	0	R/O
1	Enable	This bit controls whether the Transmit MAC state machine is active or not. When set to 0, it completes the current operation and cleanly halts. Until it is completely halted, it remains 1 when read.	0	R/W
0	Reset	When this bit is set to 1, the Transmit MAC state machine will be reset. This is a self-clearing bit.	0	R/W

Transmit MAC Status Register (Offset 0x460)

This register contains the status of the transmit Ethernet interface. Once the interface is initialized, this register is used to determine the cause of a transmit error event. Bits 4 and 5 are ORed together, and an attention is generated if the attention enable is set.

Table 266: Transmit MAC Status Register (Offset 0x460)

Bit	Field	Description	Init	Access
31-6	Reserved	-	0	R/O
5	ODI Overrun	Output Data Interface has overrun.	0	W2C
4	ODI Underrun	Output Data Interface has underrun.	0	W2C
3	Link Up	Link is up, if set.	0	R/O
2	Sent XON	An XON flow control frame was sent.	0	W2C
1	Sent XOFF	An XOFF flow control frame was sent.	0	W2C
0	RX Currently XOFFed	Received stopped due to flow control.	0	R/O

Transmit MAC Lengths Register (Offset 0x464)

This register contains various length fields that control the operation of the transmit MAC.

Table 267: Transmit MAC Lengths Register (Offset 0x464)

Bit	Field	Description	Init	Access
31-14	Reserved	-	0	R/O
13-12	IPG CRS Length	When multiplied by 2, this field indicates the number of bytes from the end of the interpacket gap (IPG) during which incoming carrier is ignored. The IEEE recommends ignoring carrier during the last 1/3 of the IPG.	00	R/W
11-8	IPG Length	When multiplied by 2, this field indicates the number of bytes in the entire IPG.	0h	R/W
7-0	Slot Time Length	When multiplied by 2, this field indicates the number of bytes in the slot time.	00h	R/W

Receive MAC Mode Register (Offset 0x468)

This register controls the receive Ethernet interfaces.

Table 268: Receive MAC Mode Register (Offset 0x468)

Bit	Field	Description	Init	Access
31-13	Reserved	-	0	R/O
12	Reserved (BCM5700 and BCM5701 only)	-	0	R/O
	Extended Hash En	Enable extended hash table size of 256 entries. By default, the hash table supports 128 entries with a 7-bit CRC value. This bit provides BCM5700 and BCM5701 legacy support by default.	0	R/W
11	Reserved (BCM5700 and BCM5701 only)	-	0	R/O
	Filt_broadcast	When set, reception of broadcast frames is disabled.	0	R/W
10	Keep VLAN Tag Diag Mode	If set, forces Receive MAC to keep the VLAN tag in the frame. This is for debugging purpose only and should be reset during normal operation.	0	R/W
9	No_CRC_Check	No CRC check by receive MAC on incoming frames. Also, allows the reception of packets received with RXERR on the MII/GMII.	0	R/W
8	Promiscuous Mode	No source address or MC hashing checking will be performed on incoming frames. All frames will be accepted.	0	R/W
7	Length Check	If set, 802.2 length checking is done on LLC frames.	0	R/W
6	Accept Runts	If set, the MAC accepts packets less than 64 bytes.	0	R/W
5	Reserved (BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	-	0	R/W
	Accept Oversized (other devices)	If set, the MAC accepts packets larger than specified in the MTU (up to 64k bytes).	0	R/W
4	Keep Pause	If set, the MAC forwards pause frame to host buffer.	0	R/W

Table 268: Receive MAC Mode Register (Offset 0x468) (Cont.)

Bit	Field	Description	Init	Access
3	Reserved	-	0	R/O
2	Enable Flow Control	Enable automatic processing of 802.3x flow control frames. This bit is orthogonal to the Keep Pause bit.	0	R/W
1	Enable	This bit controls whether the Receive MAC state machine is active or not. When set to 0, it completes the current operation and cleanly halts. Until it is completely halted, it remains 1 when read.	0	R/W
0	Reset	When this bit is set to 1, the Receive MAC state machine will be reset. This is a self-clearing bit.	0	R/W

RECEIVE MAC STATUS REGISTER (OFFSET 0X46C)

This register contains the status of the receive Ethernet interface. Once the interface is initialized, this register is used to determine the cause of a receive error event.

Table 269: Receive MAC Status Register (Offset 0x46C)

Bit	Field	Description	Init	Access
31-4	Reserved	-	000000h	R/O
3	RX FIFO Overrun	The RX FIFO has encountered an overrun condition.	0	W2C
2	XON received	A MAC Control frame with the PAUSE opcode was received with the PAUSE TIME field set to zero. The bit is sticky and must be written to clear.	0	W2C
1	XOFF received	A MAC Control frame with the PAUSE opcode was received with the PAUSE TIME field set to nonzero. The bit is sticky and must be written to clear.	0	W2C
0	Remote Transmitter XOFFed	A previously received XOFF timer has not expired yet.	0	R/O

MAC HASH REGISTER 0-3 (OFFSET 0X470-0X47C)

The hash value provides a way for the MAC to accept multicast frames through a hashing function. If the CRC hash of a multicast destination address matches a bit in the hash registers, the frame is accepted (see "Packet Filtering" on page 168).

Note: The BCM5702 MAC Transceiver and later support 128 additional hash table registers (see "MAC Hash Table Registers (Offset 0x520-0x52f)" on page 395).

Table 270: MAC Hash Register 0-3 (Offset 0x470)

Bit	Field	Description	Init	Access
Address Offset MAC Hash Register 0: 0x470				
Address Offset MAC Hash Register 1: 0x474				
Address Offset MAC Hash Register 2: 0x478				
Address Offset MAC Hash Register 3: 0x47C				
31-0	Hash Value	Hash Value for multicast destination address matching.	0	R/W

Receive Rules Control Registers (Offset Rule N: 0x480 + 8*N)

The BCM5700, BCM5701, BCM5702, BCM5703, BCM5704, BCM5714, and BCM5715 controllers implement 16 receive rules (N = 0 to 15). The BCM5705, BCM5721, and BCM5751 controllers implement eight receive rules (N = 0 to 7).

Table 271: Receive Rules Control Register (Offset 0x480)

Bit	Field	Description	Init	Access
31	Enable	Corresponding Rule is enabled when set.	0	R/W
30	And With Next	This rule and next must both be true to match. The class fields must be the same. A disabled next rule is considered true. Processor activation bits are specified in the first rule in a series.	0	R/W
29	Activate Processor 1	If the rule matches, the processor is activated in the queue descriptor for the Receive Queue Placement state machine.	0	R/W
28	Reserved (BCM5705, BCM5788, BCM5721, BCM5751, and BCM5752 only)	-	0	R/O
	Activate Processor 2 (other devices)	If the rule matches, the processor is activated in the queue descriptor for the Receive Data and Receive BD Initiator state machine.	0	R/W
27	Reserved (BCM5705, BCM5788, BCM5721, BCM5751, and BCM5752 only)	-	0	R/O
	Activate Processor 3 (other devices)	If the rule matches, the processor is activated in the queue descriptor for the Receive Data Completion state machine	0	R/W
26	Mask	If set, specifies that the value/mask field is split into a 16-bit value and 16-bit mask instead of a 32-bit value.	0	R/W
25	Discard	Discard Frame if it matches the rule.	0	R/W
24	Map	Use the masked value and map it to the class.	0	R/W
17-16	Comparison Operator	Specifies how to determine the match: 00: Equal. 01: Not equal. 10: Greater than. 11: Less than.	00	R/W
15-13	Header Type	Specifies which header the offset is for: 000: Start of Frame (always valid). 001: Start of IP Header (if present). 010: Start of TCP Header (if present). 011: Start of UDP Header (if present). 100: Start of Data (always valid, context sensitive).	000	R/W
12-8	Class	The class this frame is placed into if the rule matches. 0-16 where 0 means discard. The number of valid classes is the Number of Active Queues divided by the Number of Interrupt Distribution Groups. Ring 1 has the highest priority and ring 16 has the lowest priority.	0	R/W
7-0	Offset	Number of bytes offset specified by the header type.	0	R/W

Receive Rules Value/Mask Registers (Offset Rule N: 0x484 + 8*N)

The BCM5700, BCM5701, BCM5702, BCM5703, BCM5704, BCM5714, and BCM5715 MAC controllers implement 16 receive rules (N = 0 to 15). The BCM5705,BCM5788, BCM5721,BCM5751, and BCM5752 MAC controllers implement eight receive rules (N = 0 to 7). This register is either a 32-bit left justified value, or a 16-bit mask followed by a 16-bit value. The use of the field is determined by the mask bit of the corresponding rule.

Table 272: Receive Rules Value/Mask Register (Offset 0x484)

Bit	Field	Description	Init	Access
31-16	Mask/Value	For each bit set, the corresponding bit in the Value field is ignored during the rule match process. If bit 26 of the corresponding rule control register is set, the field is used as an additional 16-bit value for rule comparison.	0	R/W
15-0	Value	This field specifies a 16-bit value for rules comparison.	0	R/W

Receive Rules Configuration Register (Offset 0x500)

Table 273: Receive Rules Configuration Register (Offset 0x500)

Bit	Field	Description	Init	Access
31-8	Reserved	Reserved.	0	R/O
7-3	No Rules Matches Default Class	Specifies the default class of service for the frame if no rules are matched. A value of 1 is the highest priority and a value of 16 is the lowest priority. A value of zero will cause the frame to be discarded.	0	R/W
2-0	Reserved	Reserved.	0	R/O

Low Watermark Maximum Receive Frames Register (Offset 0x504)

This register is useful for flow control to prevent dropped packets.

Table 274: Low Watermark Maximum Receive Frames Register (Offset 0x504)

Bit	Field	Description	Init	Access
31-16	Reserved	Reserved.	0	R/O
15-0	Low Watermark Max Receive Frames	Specifies the number of good frames to receive after RX MBUF Low Watermark has been reached. After the RX MAC receives this number of frames, it will drop subsequent incoming frames until the MBUF High Watermark is reached. Default to zero (i.e., drop frames once RX MBUF Low Watermark is reached).	0	R/W

Note: The BCM57XX family generates a pause frame when the Low Watermark Max Receive Frames value has been reached. When a pause frame has been generated, the TX data path also stalls (assuming it has not already reached the value specified by the "Read DMA MBUF Low Watermark Register (Offset 0x4410)" on page 469), even though no pause frame was received from the Ethernet link partner.

MAC HASH TABLE REGISTERS (OFFSET 0X520-0X52F)

These registers are not applicable to the BCM5700, BCM5701, BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 devices.

The hash table can be increased to 256 indexes when the Receive MAC Mode register Extended Hash Enable bit (see "Receive MAC Mode Register (Offset 0x468)" on page 391) is set.

ETHERNET MAC PERFECT ADDRESS REGISTERS (OFFSET 0X530-0X58F)

These registers are not applicable to the BCM5700, BCM5701, BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 devices.

Table 275: Ethernet MAC Perfect Address Registers (Offset 0x530-0x58F)

Address	Description
0x530-0x537	MAC Perfect Address 4
0x538-0x53F	MAC Perfect Address 5
0x540-0x547	MAC Perfect Address 6
0x548-0x54F	MAC Perfect Address 7
0x550-0x557	MAC Perfect Address 8
0x558-0x55F	MAC Perfect Address 9
0x560-0x567	MAC Perfect Address 10
0x568-0x56F	MAC Perfect Address 11
0x570-0x577	MAC Perfect Address 12
0x578-0x57F	MAC Perfect Address 13
0x580-0x587	MAC Perfect Address 14
0x588-0x58f	MAC Perfect Address 15

SerDes Registers

SerDes Control Register (Offset 0x590)

The below definition is applicable to BCM5703S only.

Table 276: SerDes Control Register (Offset 0x590, 5703S Only)

Bit	Field	Description	Init	Access
31-24	Reserved	-	0	R/O
23-22	Regctl	Control regulator voltages (MAC1 only). Possible values include: • 00 = 2.6V (default) • 01 = 2.4V • 10 = 2.5V • 11 = 2.7V **Note:** Although the power-on default is 2.6V, software needs to program the value to 2.5V.	00	R/W
21-20	Regctl12	Controls 1.2V regulator voltages (MAC1 only). Possible values include: • 00 = 1.2V (default) • 01 = 1.1V • 10 = 1.3V • 11 = 1.4V	00	R/W
19	Rev_phase	Reverse 125-MHz receive clock phase from SerDes output.	0	R/W
18	Remote_Lbk	When set, loopback from SerDes RD± through SerDes and back to TD± is enabled.	0	R/W
17	TBI_LBK	When set, loopback from transmit TBI to receive TBI is enabled.	0	R/W
16	CDET_EN	When set, comma detection is enabled.	1	R/W
15	plltest	When set, the SerDes PLL test mode is enabled.	0	R/W
14	Serdes_MODE	When set, TXCP/TXCN are disabled.	1	R/W
13	TXedge	When clear, transmit TBI data is captured on the falling edge of internal TXWCLK. When set, data is captured on the rising edge.	1	R/W
12	TXMODE	When set, the clock recovered from the receiver is driven on the TXCP/TXCN pins	0	R/W
11	BGMIN	When set, the bias current is set to -25%	0	R/W
10	BGMax	When set, the bias current is set to +25%.	0	R/W
9-7	TXBIAS	Configures the TXDAC output swing: • 000 = Minimum • 111 = Maximum	000	R/W
6	RXCKSEL	When clear, the receive data on the TBI is available on the rising edge of internal RXWCLK. When set, data is available on the falling edge.	0	R/W
5	Reserved	-	0	R/O
4-3	RXG	Receiver Gain setting.	0	R/W
2-0	RXR	Phase interpolator bias setting.	0	R/W

The below definition is applicable to BCM5704C only.

Table 277: SerDes Control Register (Offset 0x590, BCM5704C)

Bit	Field	Description	Init	Access
31-28	Reserved	-	0	R/O
27-24	27: TX Reverse Phase 2	TX reverse phase from serdes_if to SerDes (Rev. A1 and above); use the default value all the time.	1	R/W
	26: TX Reverse Phase 1	TX reverse phase from core to serdes_if (Rev. A1 and above); use the default value all the time.	1	R/W
	25: RX Reverse Phase 1	RX reverse phase from serdes_if to EMAC (Rev. A1 and above); use the default value all the time.	0	R/W
	24: RX Reverse Phase 2	RX reverse phase from SerDes to serdes_if (Rev. A1 and above); use the default value all the time.	0	R/W
	Reserved	-	0	R/O
23-22	Regctl	Control regulator voltages (MAC1 only). Possible values include: • 00 = 2.6V (default) • 01 = 2.4V • 10 = 2.5V • 11 = 2.7V ***Note:*** Although the power-on default is 2.6V, software needs to program the value to 2.5V.	00	R/W
21-20	Regctl12	Controls 1.2V regulator voltages (MAC1 only). Possible values include: • 00 = 1.2V (default) • 01 = 1.1V • 10 = 1.3V • 11 = 1.4V	00	R/W
19-0	Reserved	-	0	R/O

The below definition is applicable to BCM5704S only.

Table 278: SerDes Control Register (Offset 0x590, 5704S Only)

Bit	Field	Description	Init	Access
31-28	Reserved	-	0	R/O
27-24	27: TX Reverse Phase 2	TX reverse phase from serdes_if to SerDes (Rev. A1 and above); use the default value all the time.	1	R/W
	26: TX Reverse Phase 1	TX reverse phase from core to serdes_if (Rev. A1 and above); use the default value all the time.	1	R/W
	25: RX reverse phase 1	RX reverse phase from serdes_if to EMAC (Rev. A1 and above); use the default value all the time.	0	R/W
	24: RX reverse phase 2	RX reverse phase from SerDes to serdes_if (Rev. A1 and above); use the default value all the time.	0	R/W

Table 278: SerDes Control Register (Offset 0x590, 5704S Only) (Cont.)

Bit	Field	Description	Init	Access
23-22	Regctl	Control regulator voltages (MAC1 only). Possible values include: • 00 = 2.6V (default) • 01 = 2.4V • 10 = 2.5V • 11 = 2.7V ***Note:*** Although the power-on default is 2.6V, software needs to program the value to 2.5V.	00	R/W
21-20	Regctl12	Controls 1.2V regulator voltages (MAC1 only). Possible values include: • 00 = 1.2V (default) • 01 = 1.1V • 10 = 1.3V • 11 = 1.4V	00	R/W
19	Reserved	-	0	R/O
18	Remove Loopback	Remove loopback for serdes_if.	0	R/W
17	TBI_LBK	When set, loopback from transmit TBI to receive TBI is enabled	0	R/W
16	CDET_EN	When set, comma detection is enabled.	1	R/W
15	Powerdown	Transmitter power down.	0	R/W
14	Pre-Emphasis Enable	Pre-emphasis enable.	0	R/W
13	TX Driver Loopback	TX driver loopback.	0	R/W
12	Load Clock Edge Select	Parallel load clock edge select.	0	R/W
11-8	Predriver Current	Predriver current (for hardware debug only). Use the default of 675 mVpp only.	0x3	R/W
7-4	TX Driver Current	TX driver current (for hardware debug only). Use the default of 675 mVpp only.	0x3	R/W
3-0	Pre-emphasis Coefficient	Pre-emphasis coefficient (for hardware debug only).	0x0	R/W

The below definition is applicable to BCM5705 and BCM5788 only.

Table 279: SerDes Control Register (Offset 0x590, BCM5705 and BCM5788 Only)

Bit	*Field*	*Description*	*Init*	*Access*
31-24	Reserved	-	0	R/O
23-22	Regctl 2.5V	Control regulator voltages (MAC1 only). Possible values include: • 00 = 2.6V (default) • 01 = 2.4V • 10 = 2.5V • 11 = 2.7V ***Note:*** Although the power-on default is 2.6V, software needs to program the value to 2.5V.	00b	R/W
21-20	Regctl 1.2V	Controls 1.2V regulator voltages (MAC1 only). Possible values include: • 00 = 1.2V (default) • 01 = 1.1V • 10 = 1.3V • 11 = 1.4V	00b	R/W
19-0	Reserved	-	0	R/O

The below definition is applicable to BCM5721, BCM5751, and BCM5752 only.

Table 280: SerDes Control Register (Offset 0x590, BCM5721, BCM5751, and BCM5752 Only)

Bit	*Field*	*Description*	*Init*	*Access*
31-28	Reserved	-	0	R/O
27-24	Regctl 2.5V	Output voltage trim control. • 0 = +0% • 1 = +2% • 2 = +4% • 3 = +6% • 4 = +8% • 5 = +10% • 6 = +12% • 7 = +14% • 8 = –16% • 9 = –14% • 10 = –12% • 11 = –10% • 12 = –8% • 13 = –6% • 14 = –4% • 15 = –2%	0 1 (for C0)	R/W

Table 280: SerDes Control Register (Offset 0x590, BCM5721, BCM5751, and BCM5752 Only) (Cont.)

Bit	Field	Description	Init	Access
23-20	Regctl 1.2V	Output voltage trim control. • 0 = +0% • 1 = +2% • 2 = +4% • 3 = +6% • 4 = +8% • 5 = +10% • 6 = +12% • 7 = +14% • 8 = –16% • 9 = –14% • 10 = –12% • 11 = –10% • 12 = –8% • 13 = –6% • 14 = –4% • 15 = –2%	1110b for A0. 0 otherwise	R/W
19-0	Reserved	-	0	R/O

SERDES STATUS REGISTER (OFFSET 0X594)

The below definition is applicable to BCM5703S MAC Transceiver SerDes only.

Table 281: SerDes Status Register (Offset 0x594, BCM5703S Only)

Bit	Field	Description	Init	Access
31-9	Reserved	-	0	R/O
8	Comma_Det	Indicates that comma code is detected.	0	R/O
7-0	RXSTAT	Receiver Status bits.	Xx	R/O

SerDes Receive Control Register (Offset 0x594)

This below register definition is applicable to BCM5704S MAC Transceiver SerDes only.

Table 282: SerDes Receive Control Register (Offset 0x594, BCM5704S Only)

Bit	Field	Description	Init	Access
31-16	Reserved	-	0	R/O
15	Power Down	Receiver power down (active high).	0	R/W
14	Bias Control 1	Bias generator control (refh).	0	R/W
13	Bias Control 2	Bias generator control (refl).	0	R/W
12	Clock Phase Load Enable	Receiver clock phase load enable (active high).	0	R/W
11	Clock Edge Select	Receiver clock edge select (0 = data synchronized with rising edge of rxwclk).	0	R/W
10-9	Loop Bandwidth Control	Loop bandwidth control (01 = 1/7, 11 = 1/63).	10	R/W
8-6	Comparator Current	Comparator current (for hardware diagnostics only).	0	R/W
5-3	Interpolator Current	Interpolator current (for hardware diagnostics only).	0	R/W
2-0	Clock Buffer Current	Clock buffer current (for hardware diagnostics only).	0	R/W

SerDes Phase Control Register (Offset 0x598)

This register is applicable to the BCM5704S MAC Transceiver SerDes only

Table 283: SerDes Phase Control Register (Offset 0x598, BCM5704S Only)

Bit	Field	Description	Init	Access
31-16	Reserved	-	0	R/O
15-0	Receiver Clock Phase Control	Receiver clock phase control (for hardware debug only).	0x3FFF	R/W

SerDes PLL Control Register (Offset 0x59C)

This register is applicable to the BCM5704S MAC Transceiver SerDes only.

Table 284: SerDes PLL Control Register (Offset 0x59C, BCM5704S Only)

Bit	Field	Description	Init	Access
31-7	Reserved		0	R/O
6	PLL Test Pad Enable	PLL test pad enable (for hardware diagnostics on port 1 only).	0	R/W
5	PLL Test Pad Port Select	PLL test pad port select (for hardware diagnostics on port 1 only).	0	R/W
4	PLL Bypass Enable	PLL bypass enable (for hardware diagnostics on port 1 only).	0	R/W
3	PLL Bypass Clock Input	PLL bypass clock input (for hardware diagnostics on port 1 only).	0	R/W
2	PLL TestPLL Enable	PLL test PLL enable (for hardware diagnostics on port 1 only).	0	R/W
1	PLL Reset	PLL reset (for hardware diagnostics on port 1 only).	0	R/W
0	PLL Power Down	PLL power down (for hardware diagnostics on port 1 only).	0	R/W

SerDes Phase Status Register (Offset 0x5A0)

This register is applicable to the BCM5704S MAC Transceiver SerDes only.

Table 285: SerDes Phase Status Register (Offset 0x5A0, BCM5704S Only)

Bit	Field	Description	Init	Access
31-18	Reserved	-	0	R/O
17	Transmit Data Status	Transmit data (sigdet) status from hardware auto-negotiation (Rev. A1 and above).	0	R/O
16	Comma Detect Status	Comma detect status from serdes_if.	0	R/O
15-0	Receiver Clock Phase Status	Receiver clock phase status.	0	R/O

Hardware Auto-Negotiation Control Register (Offset 0x5B0)

The below register definition is applicable to the BCM5704S MAC Transceiver SerDes only.

Table 286: Hardware Auto-Negotiation Control Register (Offset 0x5B0, BCM5704S Only)

Bit	Field	Description	Init	Access
31	Autoneg Enable	When set, hardware auto-negotiation is enabled.	0	R/W
30	Soft Reset	When set, a soft reset is performed.	0	R/W
29	Disable Link Ready	When set, disables link ready (xmit_data) check from hardware auto-negotiation.	0	R/W
28-25	Reserved	-	0x0	R/O
24	CRC16 Clear	Clear CRC16 bus.	1	R/W
23	EN10B	EN10B	0	R/W
22	Clear Status	When set, clears hardware auto-negotiation status.	0	R/W
21	Duplex Status	Local duplex status.	1	R/W
20	Link Status	Local link status.	1	R/W
19-18	Speed Status	Speed status.	10	R/W
17	Jumbo Packet Disable	When set, disables jumbo packets.	0	R/W
16	Autoneg Restart	When set, restarts hardware auto-negotiation.	0	R/W
15	Fiber Mode	Fiber mode.	1	R/W
14-13	Remote Fault	Remote fault.	00	R/W
12	Asymmetric Pause Capable	When set, indicates that the local NIC is asymmetric pause capable.	0	R/W
11	Pause Capable	When set, indicates that the local NIC is pause capable.	0	R/W
10	GBIC Enable	Enable GBIC.	1	R/W
9	Check End Enable	Enable check end in half-duplex mode.	0	R/W
8	Error Timer Enable	Enable error timer for auto-negotiation in SGMII (not used in the BCM5704).	0	R/W
7	Half Cycle Clock Phase	Half cycle clock phase select of sg_txclk_125.	0	R/W
6	GMII Input Select	GMII input select from pad or from internal (not used in BCM5704).	0	R/W
5	Multiplexed Status Control	Select crc16_bus or mr_adv_ability (see "Hardware Auto-Negotiation Status Register (Offset 0x5B4)" on page 404). • 1 = SGMII CRC16 Bus • 0 = MR_ADV_Ability	0	R/W
4	Comma Detect Enable	When set, comma detection is enabled.	0	R/W
3	AN Reduce Timer Enable	Enable reduce timer of auto-negotiation.	0	R/W
2	Auto-negotiation Low Enable	Enable auto-negotiation Low Enable.	0	R/W
1	Remote Loopback Test	Remote loopback for testability.	0	R/W
0	Loopback Test	Loopback test.	0	R/W

SerDes Receive Control Register (0x5B0, BCM5714 and BCM5715 only)

Table 287: SerDes Receive Control Register (0x5B0, BCM5714 and BCM5715 Only)

Bits	Field	Description	Init	Access
31:10	Reserved	-	-	R/O
9	Loopback Enable	-	0	R/W
8:0	Reserved	-	0	R/O

Hardware Auto-Negotiation Status Register (Offset 0x5B4)

The below register definition is applicable to BCM5703S and BCM5704S devices only.

Table 288: Hardware Auto-Negotiation Status Register (Offset 0x5B4)

Bit	Field	Description	Init	Access
31-23	CRC	Bit[15:7] of crc16 bus.	0x00	R/O
22-21	Multiplexed Status 1	• Link partner remote fault when Multiplexed Status Control bit of "Hardware Auto-Negotiation Control Register (Offset 0x5B0)" on page 403 = 0. • crc16 bus(6:5) when Multiplexed Status Control bit of "Hardware Auto-Negotiation Control Register (Offset 0x5B0)" on page 403 = 1.	1	R/O
20-19	Multiplexed Status 2	• Link partner pause bits when Multiplexed Status Control bit of "Hardware Auto-Negotiation Control Register (Offset 0x5B0)" on page 403 = 0. • crc16(4:3) when Multiplexed Status Control bit of "Hardware Auto-Negotiation Control Register (Offset 0x5B0)" on page 403 = 1.	0	R/O
18	Multiplexed Status 3	• Link partner half-duplex when Multiplexed Status Control bit of "Hardware Auto-Negotiation Control Register (Offset 0x5B0)" on page 403 = 0. • crc16(2) when Multiplexed Status Control bit of "Hardware Auto-Negotiation Control Register (Offset 0x5B0)" on page 403 = 1.	0	R/O
17	Multiplexed Status 4	• Link partner full-duplex when Multiplexed Status Control bit of "Hardware Auto-Negotiation Control Register (Offset 0x5B0)" on page 403 = 0. • crc16(1) when Multiplexed Status Control bit of "Hardware Auto-Negotiation Control Register (Offset 0x5B0)" on page 403 = 1.	0	R/O
16	Multiplexed Status 5	• Link partner next page when Multiplexed Status Control bit of "Hardware Auto-Negotiation Control Register (Offset 0x5B0)" on page 403 = 0. • crc16(0) when Multiplexed Status Control bit of "Hardware Auto-Negotiation Control Register (Offset 0x5B0)" on page 403 = 1.	0	R/O
15-12	Reserved	-	0	R/O
11-4	AN States	Auto-negotiation state machine states (hardware debug use only).	0	R/O
3	Comma Detector	Indicate comma detector.	0	R/O
2	MAC ACK Status	Indicate MAC acknowledge bit status.	0	R/O

Table 288: Hardware Auto-Negotiation Status Register (Offset 0x5B4) (Cont.)

Bit	Field	Description	Init	Access
1	AN Complete	Auto-negotiation complete status (1 = complete).	0	R/O
0	AN Error	Auto-negotiation error status (1 = error).	0	R/O

SerDes Transmit Control Register (0x5B4)

The below register definition is applicable to BCM5714 and BCM5715 devices only.

Table 289: SerDes Transmit Control Register (0x5B4, BCM5714 and BCM5715 Only)

Bits	Field	Description	Init	Access
31:28	Reserved	-	0	R/O
27	Remote loopback en	-	0	R/W
26	Tbi_loopback_en	-	0	R/W
25:16	Reserved	Reserved	0	R/W
15:3	Reserved	-	0	R/O
2	Loopback Enable	-	0	R/W
1:0	Reserved	-	0	R/O

UMP Multicast Match Address Mask Register (0x5E4)

Table 290: UMP Multicast Match Address Mask Register (0x5E4, BCM5714 Only)

Bits	Field	Description	Init	Access
7:0	UMP Multicast Mask	Mask a bit of UMP Multicast Match Address [47:40] during Compare. Bit 7: Masks bit-47 Bit 6: Masks bit-46 Bit 5: Masks bit-45 Bit 4: Masks bit-44 Bit 3: Masks bit-43 Bit 2: Masks bit-42 Bit 1: Masks bit-41 Bit 0: Masks bit-40	0X00	R/W

UMP VLAN MATCH REGISTER (0X5E8)

Table 291: UMP VLAN Match Register (0x5E8, BCM5714 and BCM5715 Only)

Bits	Field	Description	Init	Access
11:0	UMP Match VLAN	VLAN ID value to be matched on an incoming packet to generate RDI CPU attention.	0X00	R/W

UMP EMAC CONTROL REGISTER (0X5F0)

Table 292: UMP EMAC Control Register (0x5F0, BCM5714 and BCM5715 Only)

Bits	Field	Description	Init	Access
0	Ump_vlan_match_en	Enable UMP VLAN ID match	0	R/W
1	Ump_bcast_match_en	Enable Broadcast address match	0	R/W
2	Ump_mcast_match_en	Enable Multicast address match	0	R/W
3	Ump_mac0_match_en	Enable MAC0 address match	0	R/W
4	Ump_mac1_match_en	Enable MAC1 address match	0	R/W
5	Ump_mac2_match_en	Enable MAC2 address match	0	R/W
6	Ump_mac3_match_en	Enable MAC3 address match	0	R/W
7	Disable_mc_hash	Disables the MC hash function in L2 address filter of the MAC.	0	R/W

STATISTICS REGISTERS

The following statistics registers are applicable only to BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714C, BCM5714S, BCM5715C, and BCM5715S devices.

Table 293: Statistics Registers

Address	***Length***	***Description***
0x0800-0x0803	bits 0:27	TX MAC Statistic Counter—ifHCOutOctets
0x0804-0x0807		Reserved
0x0808-0x080b	bits 0:16	TX MAC Statistic Counter—etherStatsCollisions
0x080c-0x080f	bits 0:16	TX MAC Statistic Counter—outXonSent
0x0810-0x0813	bits 0:16	TX MAC Statistic Counter—outXoffSent
0x0814-0x0817		Reserved
0x0818-0x081b	bits 0:16	TX MAC Statistic Counter—dot3StatsInternalMacTransmitErrors
0x081c-0x081f	bits 0:16	TX MAC Statistic Counter—dot3StatsSingleCollisionFrames
0x0820-0x0823	bits 0:16	TX MAC Statistic Counter—dot3StatsMultipleCollisionFrames
0x0824-0x0827	bits 0:16	TX MAC Statistic Counter—dot3StatsDeferredTransmissions
0x0828-0x082b		Reserved
0x082c-0x082f	bits 0:16	TX MAC Statistic Counter—dot3StatsExcessiveCollisions
0x0830-0x0833	bits 0:16	TX MAC Statistic Counter—dot3StatsLateCollisions
0x0834-0x086b		Reserved
0x086c-0x086f	bits 0:27	TX MAC Statistic Counter—ifHCOutUcastPkts
0x0870-0x0873	bits 0:27	TX MAC Statistic Counter—ifHCOutMulticastPkts
0x0874-0x0877	bits 0:27	TX MAC Statistic Counter—ifHCOutBroadcastPkts
0x0878-0x087b	bits 0:16	TX MAC Statistic Counter—dot3StatsCarrierSenseErrors
0x087c-0x087f	bits 0:16	TX MAC Statistic Counter—ifOutDiscards
0x0880-0x0883	bits 0:31	RX MAC Statistic Counter—ifHCInOctets
0x0884-0x0887		Reserved
0x0888-0x088b	bits 0:16	RX MAC Statistic Counter—etherStatsFragments
0x088c-0x088f	bits 0:31	RX MAC Statistic Counter—ifHCInUcastPkts
0x0890-0x0893	bits 0:31	RX MAC Statistic Counter—ifHCInMulticastPkts
0x0894-0x0897	bits 0:31	RX MAC Statistic Counter—ifHCInBroadcastPkts
0x0898-0x089b	bits 0:16	RX MAC Statistic Counter—dot3StatsFCSErrors
0x089c-0x089f	bits 0:16	RX MAC Statistic Counter—dot3StatsAlignmentErrors
0x08a0-0x08a3	bits 0:16	RX MAC Statistic Counter—xonPauseFramesReceived
0x08a4-0x08a7	bits 0:16	RX MAC Statistic Counter—xoffPauseFramesReceived
0x08a8-0x08ab	bits 0:16	RX MAC Statistic Counter—macControlFramesReceived
0x08ac-0x08af	bits 0:16	RX MAC Statistic Counter—xoffStateEntered
0x08b0-0x08b3	bits 0:16	RX MAC Statistic Counter—dot3StatsFramesTooLong
0x08b4-0x08b7	bits 0:16	RX MAC Statistic Counter—etherStatsJabbers
0x08b8-0x08bb	bits 0:16	RX MAC Statistic Counter—etherStatsUndersizePkts
0x08bc-0x8ff		Reserved

Transmit MAC Statistic Counters

ifHCOutOctets (Offset 0x0800)

The number of octets transmitted out of the interface, including framing characters.

etherStatsCollisions (Offset 0x0808)

The number of collisions experienced.

outXonSent (Offset 0x080C)

Sent Xon.

outXoffSent (Offset 0x0810)

Sent Xoff.

dot3StatsInternalMacTransmitErrors (Offset 0x0818)

A count of frames for which transmission on a particular interface fails due to an internal MAC sublayer transmit error.

dot3StatsSingleCollisionFrames (Offset 0x081C)

A count of successfully transmitted frames on a particular interface for which transmission is inhibited by exactly one collision.

dot3StatsMultipleCollisionFrames (Offset 0x0820)

A count of successfully transmitted frames on a particular interface for which transmission is inhibited by more than one collision.

dot3StatsDeferredTransmissions (Offset 0x0824)

A count of frames for which the first transmission attempt on a particular interface is delayed because the medium is busy.

dot3StatsExcessiveCollisions (Offset 0x082C)

A count of frames for which transmission on a particular interface fails due to excessive collisions.

dot3StatsLateCollisions (Offset 0x0830)

The number of times that a collision is detected on a particular interface later than 512 bit-times into the transmission of a packet.

ifHCOutUcastPkts (Offset 0x086C)

The number of packets that higher-level protocols requested be transmitted, and that were not addressed to a multicast or broadcast address at this sublayer, including those that were discarded or not sent.

ifHCOutMulticastPkts (Offset 0x0870)

The number of packets that higher-level protocols requested be transmitted, and that were addressed to a multicast address at this sublayer, including those that were discarded or not sent.

ifHCOutBroadcastPkts (Offset 0x0874)

The number of packets that higher-level protocols requested be transmitted, and that were addressed to a broadcast address at this sublayer, including those that were discarded or not sent.

Receive MAC Statistic Counters

ifHCInOctets (Offset 0x0880)

The number of octets received on the interface, including framing characters.

etherStatsFragments (Offset 0x0888)

A frame size that is less than 64 bytes with a bad FCS.

ifHCInUcastPkts (Offset 0x088C)

The number of packets delivered by this sublayer to a higher (sub)layer, which were not addressed to a multicast or broadcast address at this sublayer.

ifHCInMulticastPkts (Offset 0x0890)

The number of packets delivered by this sublayer to a higher (sub)layer, which were addressed to a multicast address at this sublayer.

ifHCInBroadcastPkts (Offset 0x0894)

The number of packets delivered by this sublayer to a higher (sub)layer, which were addressed to a broadcast address at this sublayer.

dot3StatsFCSErrors (Offset 0x0898)

A count of frames received on a particular interface that are an integral number of octets in length and do not pass the FCS check.

dot3StatsAlignmentErrors (Offset 0x089C)

A count of frames received on a particular interface that are not an integral number of octets in length and do not pass the FCS check.

xonPauseFramesReceived (Offset 0x08A0)

MAC control frames with pause command and length equal to zero.

xoffPauseFramesReceived (Offset 0x08A4)

MAC control frames with pause command and length greater than zero.

macControlFramesReceived (Offset 0x08A8)

MAC control frames with no pause command.

xoffStateEntered (Offset 0x08AC)

Transmitting is disabled.

dot3StatsFramesTooLongs (Offset 0x08B0)

A count of frames received on a particular interface that exceeds the maximum permitted frame size.

etherStatsJabbers (Offset 0x08B4)

Frames exceed jabber time.

etherStatsUndersizePkts (0x08B8)

Frames with a size less than 64 bytes.

SEND DATA INITIATOR CONTROL REGISTERS

Table 294: Send Data Initiator Control Registers

Offset	Registers
0x0c00-0x0c03	Send Data Initiator Mode
0x0c04-0x0c07	Send Data Initiator Status
0x0c08-0x0c0b	Send Data Initiator Statistics Control
0x0c0c-0x0c0f	Send Data Initiator Statistics Enable Mask
0x0c10-0x0c13	Send Data Initiator Statistics Increment Mask
0x0c14-0x0c7f	Reserved
0x0c80-0x0c83	Local Statistics Counter: Class of Service 1
0x0c84-0x0c87	Local Statistics Counter: Class of Service 2
0x0c88-0x0c8b	Local Statistics Counter: Class of Service 3
0x0c8c-0x0c8f	Local Statistics Counter: Class of Service 4
0x0c90-0x0c93	Local Statistics Counter: Class of Service 5
0x0c94-0x0c97	Local Statistics Counter: Class of Service 6
0x0c98-0x0c9b	Local Statistics Counter: Class of Service 7
0x0c9c-0x0c9f	Local Statistics Counter: Class of Service 8
0x0ca0-0x0ca3	Local Statistics Counter: Class of Service 9
0x0ca4-0x0ca7	Local Statistics Counter: Class of Service 10
0x0ca8-0x0cab	Local Statistics Counter: Class of Service 11
0x0cac-0x0caf	Local Statistics Counter: Class of Service 12
0x0cb0-0x0cb3	Local Statistics Counter: Class of Service 13
0x0cb4-0x0cb7	Local Statistics Counter: Class of Service 14
0x0cb8-0x0cbb	Local Statistics Counter: Class of Service 15
0x0cbc-0x0cbf	Local Statistics Counter: Class of Service 16
0x0cc0-0x0cc3	Local Statistics Counter: DMA Read Queue Full
0x0cc4-0x0cc7	Local Statistics Counter: DMA High Priority Read Queue Full
0x0cc8-0x0ccb	Local Statistics Counter: SDC Queue Full
0x0ccc-0x0ccf	Local Statistics Counter: NIC Ring Set Send Producer Index
0x0cd0-0x0cd3	Local Statistics Counter: Status Updated
0x0cd4-0x0cd7	Local Statistics Counter: Interrupts
0x0cd8-0x0cdb	Local Statistics Counter: Avoided Interrupts
0x0cdc-0x0cdf	Local Statistics Counter: Send Threshold Hit
0x0ce0-0x0fff	Reserved

SEND DATA INITIATOR MODE REGISTER (OFFSET 0X0C00)

Table 295: Send Data Initiator Mode Register (Offset 0x0C00)

Bit	Field	Description	Init	Access
31-6	Reserved	-	0	R/O
5	Multiple Segment Enable (BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	Enable Read DMA to read multi-segment (up to four segments) in one DMA request during TCP segmentation.	0	R/W
	Reserved (other devices)	-	0	R/O
4	Pre-DMA Debug Enable(BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	When this bit is set, the Send Data Initiator state machine will be halted if the pre-DMA bit of the Send BD is set.	0	R/W
	Reserved (other devices)	-	0	R/O
3	Hardware Pre-DMA Enable (BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	Enable hardware LSO pre-DMA processing.	0	R/W
	Reserved (other devices)	-	0	R/O
2	Stats Overflow Attn Enable	Enable attention for statistics overflow.	0	R/W
1	Enable	This bit controls whether the Send Data Initiator state machine is active or not. When set to 0, it completes the current operation and cleanly halts. Until it is completely halted, it remains one when read.	1	R/W
0	Reset	When this bit is set to 1, the Send Data Initiator state machine is reset. This is a self-clearing bit.	0	R/W

SEND DATA INITIATOR STATUS REGISTER (OFFSET 0X0C04)

Table 296: Send Data Initiator Status Register (Offset 0x0C04)

Bit	Field	Description	Init	Access
31-3	Reserved	-	0	R/O
2	Stats Overflow Attn	A statistics managed by Send Data Initiator has overflowed.	0	R/O
1-0	Reserved	-	0	R/O

SEND DATA INITIATOR STATISTICS CONTROL REGISTER (OFFSET 0X0C08)

Table 297: Send Data Initiator Statistics Control Register (Offset 0x0C08)

Bit	Field	Description	Init	Access
31-5	Reserved	-	0	R/O
4	Reserved (BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	-	0	R/O
	Force Statistics Zero (other devices)	If set, forces the statistics in the NIC memory to zeros. This should be done when statistics enable is low. Only the masked statistics will be cleared. This is a self-clearing bit.	0	R/W
3	Reserved (BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	-	0	R/O
	Force Statistics Flush (other devices)	If set, forces a flush of the local statistics to the NIC memory by doing a read-modify-write operation. This can be set only when statistics enable is low. Only the masked statistics will be flushed. Self-clearing when flush completes.	0	R/W
2	Statistics Clear	If set, resets local statistics counters to zero. Clears only masked statistics. Self-clearing when done.	0	R/W
1	Reserved (BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	-	0	R/O
	Faster Statistics Update (other devices)	Allow a faster update of the statistics counters to the NIC memory. When set, one statistics is updated every 15 us (or 998 clocks). When reset, one statistics is updated every 25 us (or 1662 clocks).	0	R/W
0	Statistics Enable	When set, allows the local statistics counters to increment. When reset, counters hold their values until next update to the NIC memory. Enables only masked statistics.	0	R/W

Send Data Initiator Statistics Enable Mask Register (Offset 0x0C0C)

This version of the Send Data Initiator Statistics Enable Mask register applies to the BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 devices only.

Table 298: Send Data Initiator Statistics Enable Mask Register (Offset 0x0C0C)

Bit	Field	Description	Init	Access
31-19	Reserved	-	0	R/O
18-16	Counters Enable Mask	Mask controls which statistics can be updated, cleared or flushed. Bits 16-18 correspond to DMA Read Queue Full, DMA High Priority Read Queue Full, and Send Data Completion Queue Full respectively.	0	R/W
15-1	Reserved	-	0	R/O
0	Counters Enable Mask	Controls whether Class of Service 0 statistics can be updated, cleared, or flushed.	0	R/W

Rest of BCM57XX Family

This version of the Send Data Initiator Statistics Enable Mask register applies to the rest of the BCM57XX family.

Table 299: Send Data Init. Stat. Enable Mask Register (Offset 0x0C0C, Rest of BCM57XX Family)

Bit	Field	Description	Init	Access
31-24	Reserved	-	0	R/O
23-0	Counters Enable Mask	Controls which statistics can be updated, cleared or flushed. • Bits 0-15 correspond to statistics for Class of Service 1-16. • Bits 16-23 correspond to DMA Read Queue Full, DMA High Priority Read Queue Full, and Send Data Completion Queue Full, Set Send Producer Index, Status Updated, Interrupts, Avoided Interrupts, Send Threshold Hit respectively.	0	R/W

SEND DATA INITIATOR STATISTICS INCREMENT MASK REGISTER (OFFSET 0X0C10)

This version of the Send Data Initiator Statistics Enable Mask register applies to the BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 devices only.

Table 300: Send Data Initiator Statistics Increment Mask Register (Offset 0x0C10)

Bit	Field	Description	Init	Access
31-24	Reserved	-	0	R/O
23-19	Counters Increment Mask	Writing a 1 to the bit position forces the corresponding statistics counter to increment by 1. Not affected by Statistics Enable Mask. Bits 16-23 correspond to Set Send Producer Index, Status Updated, Interrupts, Avoided Interrupts, Send Threshold Hit respectively.	0	W/O
18-16	Reserved	-	0	R/O
15-0		Writing a 1 to the bit position forces the corresponding statistics counter to increment by 1. Not affected by the Statistics Enable mask. Bits 15-0 correspond to statistics for Class of Service 16-1.	0	W/O

Rest of BCM57XX Family

This version of the Send Data Initiator Statistics Increment Mask register applies to the rest of the BCM57XX family.

Table 301: Send Data Init. Stat. Increment Mask Register (Offset 0x0C10, Rest of BCM57XX Family)

Bit	Field	Description	Init	Access
31-24	Reserved	-	0	R/O
23-0	Counters Increment Mask	Writing a 1 to the bit position forces the corresponding statistics counter to increment by 1. Not affected by Statistics Enable Mask. • Bits 0-15 correspond to statistics for Class of Service 1-16. • Bits 16-23 correspond to DMA Read Queue Full, DMA High Priority Read Queue Full, and Send Data Completion Queue Full, Set Send Producer Index, Status Updated, Interrupts, Avoided Interrupts, Send Threshold Hit respectively.	0	W/O

LOCAL STATISTICS COUNTERS (OFFSET 0X0C80-0X0CDF)

The registers 0xC84-0xCBF and 0x0CCC-0x0CDF are reserved in BCM5714 and BCM5715 devices.

Table 302: Local Statistics Counters (Offset 0x0C80-0x0CDF)

Bit	Field	Description	Init	Access
31-10	Reserved	-	0	R/O
9-0	Counter Value	The current counter value for statistics kept by the Send Data Initiator.		R/O

TCP SEGMENTATION CONTROL REGISTERS

These registers are applicable to the BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 devices only.

Table 303: TCP Segmentation Control Registers

Offset	Registers
0x0ce0-0x0ce3	Lower Host Address Register for TCP Segmentation
0x0ce4-0x0ce7	Upper Host Address Register for TCP Segmentation
0x0ce8-0x0ceb	Length/Offset Register for TCP Segmentation
0x0cec-0x0cef	DMA Flags Register for TCP Segmentation
0x0cf0-0x0cf3	VLAN Tag Register for TCP Segmentation
0x0cf4-0x0cf7	Pre-DMA Command Exchange Register for TCP Segmentation

LOWER HOST ADDRESS REGISTER FOR TCP SEGMENTATION (OFFSET 0XCE0)

This register is applicable to the BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 devices only.

Table 304: Lower Host Address Register for TCP Segmentation (Offset 0xCE0)

Bit	Field	Description	Init	Access
31-0		Specifies the lower 32 bits of the starting address in host memory where the transmit data buffer resides.	0	R/W

UPPER HOST ADDRESS REGISTER FOR TCP SEGMENTATION (OFFSET 0XCE4)

This register is applicable to the BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 devices only.

Table 305: Upper Host Address Register for TCP Segmentation (Offset 0xCE4)

Bit	Field	Description	Init	Access
31-0		Specifies the upper 32 bits of the starting address in host memory where the transmit data buffer resides.	0	R/W

Length/Offset Register for TCP Segmentation (Offset 0xCE8)

This register is applicable to the BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 devices only.

Table 306: Length/Offset Register for TCP Segmentation (Offset 0xCE8)

Bit	Field	Description	Init	Access
31-23	Reserved	The bits can be written/read, but has no chip impact.	0	R/O
22-16	MBUF offset	MBUF offset. It specifies the offset of the first TXMBUF at where the DMA starts putting data. The valid value is between 48 and 128.	0	R/W
15-0		Specifies the length of data to be transmitted. Although firmware can specify up to 64 KB, it should not attempt to program more than 8 KB because it would exceed the size of TXMBUF.	0	R/W

DMA Flags Register for TCP Segmentation (Offset 0xCEC)

This register is applicable to the BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 devices only.

Table 307: DMA Flags Register for TCP Segmentation (Offset 0xCEC)

Bit	Field	Description	Init	Access
31-20	Reserved	-	0	R/O
19	MBUF offset valid	MBUF offset valid. When this bit is set, the RDMA engine will DMA the data into the TXMBUF starting at an offset specified in the Length/Offset register (see "Length/Offset Register for TCP Segmentation (Offset 0xCE8)" on page 417).	0	R/W
18	Last Fragment	Last Fragment. This bit is passed transparently to the SDC. When this bit is set, the SDC will inform the HC to increment the Send Ring Consumer Index. • The bit is always set by hardware if no firmware assisted TCP segmentation occurs. • Otherwise, firmware sets it at the end of fragmentation.	0	R/W
17	No Word Swap	No Word Swap. Set to disable endian word swap on data from PCI bus.	0	R/W
16	Reserved	The bit can be written/read, but has no chip impact.	0	R/O
15-14	MAC source address insertion	MAC source address insertion. This 2-bit field determines which of the four MAC addresses should be inserted into the frame.	0	R/W
13	MAC source address insertion	MAC source address insertion. Indicates that the predetermined source address is inserted into the Ethernet header of the frame.	0	R/W
12	TCP/UDP checksum enable	TCP/UDP checksum enable.	0	R/W
11	IP checksum enable	IP checksum enable.	0	R/W

Table 307: DMA Flags Register for TCP Segmentation (Offset 0xCEC) (Cont.)

Bit	Field	Description	Init	Access
10	Force RAW checksum enable	Force RAW checksum enable.	0	R/W
9	Checksum offset	Checksum offset. • When bit 10 is set to 1 and this bit is 0, the checksum will start at offset of 14. • When bit 10 is set to 1 and this bit is 1, the checksum will start at offset of 0 (i.e., checksum calculation will be performed on all data written into TXMBUF).	0	R/O
8	Reserved	The bit can be written/read, but has no chip impact.	0	R/W
7	VLAN Tag Present	VLAN Tag Present. Indicates that the VLAN tag should be copied into the Frame Header by the DMA engine.	0	R/W
6	Force Interrupt	Force Interrupt. Following the completion of this DMA, a host interrupt is generated.	0	R/W
5	Last BD in Frame	Last BD in Frame.	0	R/W
4	Coalesce Now	Coalesce Now. Pass through Send Buffer Descriptor flag.	0	R/W
3	Reserved	The bit can be written/read, but has no chip impact.	0	R/O
2	Invoke Processor	Invoke Processor. Clear the PASS bit of the entry queued to the SDCQ, so that SDC will invoke the CPU. • If the packet is created by hardware, this bit will be the same as bit 9 (BD_FLAG_CPU_POST_DMA) of the flag field in the Send BD. • If the packet is created by firmware, it will be up to CPU whether it needs to post-process the data.	0	R/W
1	Don't Generate CRC	Do not Generate CRC. Pass through Send Buffer Descriptor flag.	0	R/W
0	No Byte Swap	No Byte Swap. Set to disable endian byte swap on data from PCI bus.	0	R/W

VLAN TAG REGISTER FOR TCP SEGMENTATION (OFFSET 0XCF0)

This register is applicable to the BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 devices only.

Table 308: VLAN Tag Register for TCP Segmentation (Offset 0xCF0)

Bit	Field	Description	Init	Access
31-16	Reserved	-	0	R/O
15-0		VLAN tag to be inserted into the Frame Header if bit 7 of DMA Flags register is set.	0	R/W

PRE-DMA COMMAND EXCHANGE REGISTER FOR TCP SEGMENTATION (OFFSET 0XCF4)

This register is applicable to the BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 devices only.

Table 309: VLAN Tag Register for TCP Segmentation (Offset 0xCF0)

Bit	Field	Description	Init	Access
31	READY	The CPU sets this bit to tell the SDI that the DMA address, length, flags, and VLAN tag are valid and the request is ready to be go. The CPU polls this bit to be clear for the completion of the request.	0	R/W
30	PASS	If this bit is set to 0, the CPU will be responsible for processing the buffer descriptor.	1	R/W
29	SKIP	The CPU sets this bit to 1 to inform the SDI that the TCP segmentation is completed, and the BD_Index can be incremented.	0	R/W
28-7	Reserved	-	0	R/O
6-0	BD_Index	The internal current buffer descriptor pointer that the hardware/firmware is servicing.	0	R/W

SEND DATA COMPLETION CONTROL REGISTERS

Table 310: Send Data Completion Control Registers

Offset	Registers
0x1000-0x1003	Send Data Completion Mode.
0x1004-0x1007	Reserved.
0x1008-0x100B	Post-DMA Command Exchange for TCP Segmentation (BCM5705, BCM5714, BCM5721, and BCM5751 only).
0x100C-0x13FF	Reserved.

SEND DATA COMPLETION MODE REGISTER (OFFSET 0X1000)

Table 311: Send Data Completion Mode Register (Offset 0x1000)

Bit	Field	Description	Init	Access
31-3	Reserved	-	0	R/O
2	Long BD Burst Read Fix (BCM5705 A2 only)	When this bit is cleared and the device is operating in Long Burst Mode (see Table 402 on page 477), there is no requirement that the consumer and producer indices of the Standard Receive Producer Ring be less than 64.	0	R/W
	Reserved	-	0	R/O
1	Enable	This bit controls whether the Send Data Completion state machine is active or not. When set to 0, it completes the current operation and cleanly halts. Until it is completely halted, it remains one when read.	1	R/W
0	Reset	When this bit is set to 1, the Send Data Completion state machine is reset. This is a self-clearing bit.	0	R/W

POST-DMA COMMAND EXCHANGE REGISTER FOR TCP SEGMENTATION (OFFSET 0X1008)

This register is applicable to BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 devices only.

Table 312: Post-DMA Command Exchange Register for TCP Segmentation (Offset 0x1008)

Bit	Field	Description	Init	Access
31	PASS	If this bit is set to 0, the CPU will be invoked to process the TXMBUF data. It is the same as SDCQ bit 13.	1	R/W
30	SKIP	The CPU sets this bit to 1 to inform the SDC that the post-processing is completed and the hardware can resume operation.	0	R/W
29	End of Fragmentation	End of Fragmentation. If this bit is set to 1, the SDC will requests the HC to increment the Send Ring Consumer Index when the CPU sets the SKIP bit. It is the same as SDCQ bit 12.	1	R/W
28-12	Reserved	-	0	R/O
11-6	Head TXMBUF Pointer	Head TXMBUF Pointer. They are the same as SDCQ bits 11:6.	0	R/W
5-0	Tail TXMBUF Pointer	Tail TXMBUF Pointer. They are the same as SDCQ bits 5:0.	0	R/W

SEND BD RING SELECTOR CONTROL REGISTERS

The following registers may be used by software for debug and diagnostic purposes. For example, Host software could compare the Send BD Consumer Index located in the Status block (see "Status Block" on page 103) to the registers located in this region.

Table 313: Send BD Ring Selector Control Registers

Offset	***Registers***
0x1400-0x1403	Send BD Ring Selector Mode
0x1404-0x1407	Send BD Ring Selector Status
0x1408-0x140b	Send BD Ring Selector Hardware Diagnostics
0x140c-0x143f	Reserved
0x1440-0x1443	Send BD Diagnostic Ring Selector Local NIC Send BD 1 Consumer Index
0x1444-0x1447	Send BD Diagnostic Ring Selector Local NIC Send BD 2 Consumer Index
0x1448-0x144b	Send BD Diagnostic Ring Selector Local NIC Send BD 3 Consumer Index
0x144c-0x144f	Send BD Diagnostic Ring Selector Local NIC Send BD 4 Consumer Index
0x1450-0x1453	Send BD Diagnostic Ring Selector Local NIC Send BD 5 Consumer Index
0x1454-0x1457	Send BD Diagnostic Ring Selector Local NIC Send BD 6 Consumer Index
0x1458-0x145b	Send BD Diagnostic Ring Selector Local NIC Send BD 7 Consumer Index
0x145c-0x145f	Send BD Diagnostic Ring Selector Local NIC Send BD 8 Consumer Index
0x1460-0x1463	Send BD Diagnostic Ring Selector Local NIC Send BD 9 Consumer Index
0x1464-0x1467	Send BD Diagnostic Ring Selector Local NIC Send BD 10 Consumer Index
0x1468-0x146b	Send BD Diagnostic Ring Selector Local NIC Send BD 11 Consumer Index
0x146c-0x146f	Send BD Diagnostic Ring Selector Local NIC Send BD 12 Consumer Index
0x1470-0x1473	Send BD Diagnostic Ring Selector Local NIC Send BD 13 Consumer Index
0x1474-0x1477	Send BD Diagnostic Ring Selector Local NIC Send BD 14 Consumer Index
0x1478-0x147b	Send BD Diagnostic Ring Selector Local NIC Send BD 15 Consumer Index
0x147c-0x147f	Send BD Diagnostic Ring Selector Local NIC Send BD 16 Consumer Index
0x1480-0x17ff	Reserved

SEND BD RING SELECTOR MODE REGISTER (OFFSET 0X1400)

Table 314: Send BD Ring Selector Mode Register (Offset 0x1400)

Bit	Field	Description	Init	Access
31-3	Reserved	-	0	R/O
2	Attn_Enable	When this bit is set to 1, an internal attention is generated when an error occurs.		R/W
1	Enable	This bit controls whether the Send BD Ring Selector state machine is active or not. When set to 0, it completes the current operation and cleanly halts. Until it is completely halted, it remains 1 when read.		R/W
0	Reset	When this bit is set to 1, the Send BD Ring Selector state machine is reset. This is a self-clearing bit.		R/W

SEND BD RING SELECTOR STATUS REGISTER (OFFSET 0X1404)

Table 315: Send BD Ring Selector Status Register (Offset 0x1404)

Bit	Field	Description	Init	Access
31-3	Reserved	-	0	R/O
2	Error	Send BD Ring Selector error status.		R/O
1-0	Reserved	-	0	R/O

SEND BD RING SELECTOR HARDWARE DIAGNOSTICS REGISTER (OFFSET 0X1408)

This version of the Send BD Ring Selector Hardware Diagnostics register applies to BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 devices only.

Table 316: Send BD Ring Selector Hardware Diagnostics Register (Offset 0x1408)

Bit	Field	Description	Init	Access
31-0	Reserved	-	0	R/O

Rest of BCM57XX Family

This version of the Send BD Ring Selector Hardware Diagnostics register applies to the rest of the BCM57XX family.

Table 317: Send BD Ring Selector HW Diag. Register (Offset 0x1408, Rest of BCM57XX Family)

Bit	Field	Description	Init	Access
31-16	Reserved	-	0	R/O
15-12	SBDS RNSTMS	Ring number sent to Mailbox block.		R/O
11-8	SBDS SRN	Send BD Ring Selector Staged Ring Number.		R/O
7-4	SBDS CRN	Current Ring Number.		R/O
3-0	SBDS State	Current Send BD Ring Selector State.		R/O

Send BD Diagnostic Ring Selector Local NIC Send BD Consumer Index Registers (Offset 0x1440-0x147C)

This set of registers is used to keep track of the current DMAs queued to move send data from the host to the NIC.

Table 318: Send BD Diag. Ring Selector Local NIC Send BD Consumer Index Registers (Offset 0x1440)

Bit	Field	Description	Init	Access
31-9	Reserved	-	0	R/O
8-0	Index	These nine bits contain the current NIC send BD index.		R/O

Send BD Initiator Control Registers

These registers are available for diagnostic and debug purposes. For example, host software may compare the value written to the high priority mailbox region (see "High-Priority Mailboxes" on page 370) against the value the MAC processes located in the Send BD Initiator Control register region.

Table 319: Send BD Ring Selector Control Registers

Offset	Registers
0x1800-0x1803	Send BD Initiator Mode
0x1804-0x1807	Send BD Initiator Status
0x1808-0x180b	Send BD Diagnostic Initiator Local NIC Send BD 1 Producer Index
0x180c-0x180f	Send BD Diagnostic Initiator Local NIC Send BD 2 Producer Index
0x1810-0x1813	Send BD Diagnostic Initiator Local NIC Send BD 3 Producer Index
0x1814-0x1817	Send BD Diagnostic Initiator Local NIC Send BD 4 Producer Index
0x1818-0x181b	Send BD Diagnostic Initiator Local NIC Send BD 5 Producer Index
0x181c-0x181f	Send BD Diagnostic Initiator Local NIC Send BD 6 Producer Index
0x1820-0x1823	Send BD Diagnostic Initiator Local NIC Send BD 7 Producer Index
0x1824-0x1827	Send BD Diagnostic Initiator Local NIC Send BD 8 Producer Index
0x1828-0x182b	Send BD Diagnostic Initiator Local NIC Send BD 9 Producer Index
0x182c-0x182f	Send BD Diagnostic Initiator Local NIC Send BD 10 Producer Index
0x1830-0x1833	Send BD Diagnostic Initiator Local NIC Send BD 11 Producer Index
0x1834-0x1837	Send BD Diagnostic Initiator Local NIC Send BD 12 Producer Index
0x1838-0x183b	Send BD Diagnostic Initiator Local NIC Send BD 13 Producer Index
0x183c-0x183f	Send BD Diagnostic Initiator Local NIC Send BD 14 Producer Index
0x1840-0x1843	Send BD Diagnostic Initiator Local NIC Send BD 15 Producer Index
0x1844-0x1847	Send BD Diagnostic Initiator Local NIC Send BD 16 Producer Index
0x1848-0x1bff	Reserved

SEND BD INITIATOR MODE REGISTER (OFFSET 0X1800)

Table 320: Send BD Initiator Mode Register (Offset 0x1800)

Bit	Field	Description	Init	Access
31-3	Reserved	-	0	R/O
2	Attn_Enable	When this bit is set to 1, an internal attention is generated when an error occurs.	-	R/W
1	Enable	This bit controls whether the Send BD Initiator state machine is active or not. When set to 0, it completes the current operation and cleanly halts. Until it is completely halted, it remains 1 when read.	1	R/W
0	Reset	When this bit is set to 1, the Send BD Initiator state machine is reset. This is a self-clearing bit.	0	R/W

SEND BD INITIATOR STATUS REGISTER (OFFSET 0X1804)

Table 321: Send BD Initiator Status Register (Offset 0x1804)

Bit	Field	Description	Init	Access
31-3	Reserved	-	0	R/O
2	Error	Sends BD Initiator error.	-	R/O
1-0	Reserved	-	0	R/O

SEND BD DIAGNOSTIC INITIATOR LOCAL NIC SEND BD *N* PRODUCER INDEX REGISTERS (OFFSET 0X1808-0X1844)

This set of registers is used to keep track of the current DMAs queued to move send BDs from the host to the NIC.

SEND BD COMPLETION CONTROL REGISTERS

Table 322: Send BD Completion Control Registers

Offset	Registers
0x1c00-0x1c03	Send BD Completion Mode.
0x1c04-0x1fff	Reserved.

SEND BD COMPLETION MODE REGISTER (OFFSET 0X1C00)

Table 323: Send BD Completion Mode Register (Offset 0x1C00)

Bit	Field	Description	Init	Access
31-3	Reserved	-	0	R/O
2	Attn_Enable	When this bit is set to 1, an internal attention is generated when an error occurs.	0	R/W
1	Enable	This bit controls whether the Send BD Completion state machine is active or not. When set to 0, it completes the current operation and cleanly halts. Until it is completely halted, it remains one when read.	1	R/W
0	Reset	When this bit is set to 1, the Send BD Completion state machine is reset. This is a self-clearing bit.	0	R/W

RECEIVE LIST PLACEMENT CONTROL REGISTERS

Table 324: Receive List Placement Control Registers

Offset	Registers
0x2000-0x2003	Receive List Placement Mode
0x2004-0x2007	Receive List Placement Status
0x2008-0x200b	Receive Selector List Lock Register
0x200c-0x200f	Receive Selector Non-Empty Bits Register
0x2010-0x2013	Receive List Placement Configuration Register
0x2014-0x2017	Receive List Placement Statistics Control
0x2018-0x201b	Receive List Placement Statistics Enable Mask
0x201c-0x201f	Receive List Placement Statistics Increment Mask
0x2020-0x20ff	Reserved
0x2100-0x2103	Receive Selector List 1 Head
0x2104-0x2107	Receive Selector List 1 Tail
0x2108-0x210b	Receive Selector List 1 Count
0x210c-0x210f	Reserved
0x2110-0x2113	Receive Selector List 2 Head
0x2114-0x2117	Receive Selector List 2 Tail
0x2118-0x211b	Receive Selector List 2 Count
0x211c-0x211f	Reserved
0x2120-0x2123	Receive Selector List 3 Head
0x2124-0x2127	Receive Selector List 3 Tail
0x2128-0x212b	Receive Selector List 3 Count
0x212c-0x212f	Reserved
0x2130-0x2133	Receive Selector List 4 Head
0x2134-0x2137	Receive Selector List 4 Tail
0x2138-0x213b	Receive Selector List 4 Count
0x213c-0x213f	Reserved
0x2140-0x2143	Receive Selector List 5 Head
0x2144-0x2147	Receive Selector List 5 Tail
0x2148-0x214b	Receive Selector List 5 Count
0x214c-0x214f	Reserved
0x2150-0x2153	Receive Selector List 6 Head
0x2154-0x2157	Receive Selector List 6 Tail
0x2158-0x215b	Receive Selector List 6 Count
0x215c-0x215f	Reserved
0x2160-0x2163	Receive Selector List 7 Head
0x2164-0x2167	Receive Selector List 7 Tail

Table 324: Receive List Placement Control Registers (Cont.)

Offset	***Registers***
0x2168-0x216b	Receive Selector List 7 Count
0x216c-0x216f	Reserved
0x2170-0x2173	Receive Selector List 8 Head
0x2174-0x2177	Receive Selector List 8 Tail
0x2178-0x217b	Receive Selector List 8 Count
0x217c-0x217f	Reserved
0x2180-0x2183	Receive Selector List 9 Head
0x2184-0x2187	Receive Selector List 9 Tail
0x2188-0x218b	Receive Selector List 9 Count
0x218c-0x218f	Reserved
0x2190-0x2193	Receive Selector List 10 Head
0x2194-0x2197	Receive Selector List 10 Tail
0x2198-0x219b	Receive Selector List 10 Count
0x219c-0x219f	Reserved
0x21a0-0x21a3	Receive Selector List 11 Head
0x21a4-0x21a7	Receive Selector List 11 Tail
0x21a8-0x21ab	Receive Selector List 11 Count
0x21ac-0x21af	Reserved
0x21b0-0x21b3	Receive Selector List 12 Head
0x21b4-0x21b7	Receive Selector List 12 Tail
0x21b8-0x21bb	Receive Selector List 12 Count
0x21bc-0x21bf	Reserved
0x21c0-0x21c3	Receive Selector List 13 Head
0x21c4-0x21c7	Receive Selector List 13 Tail
0x21c8-0x21cb	Receive Selector List 13 Count
0x21cc-0x21cf	Reserved
0x21d0-0x21d3	Receive Selector List 14 Head
0x21d4-0x21d7	Receive Selector List 14 Tail
0x21d8-0x21db	Receive Selector List 14 Count
0x21dc-0x21df	Reserved
0x21e0-0x21e3	Receive Selector List 15 Head
0x21e4-0x21e7	Receive Selector List 15 Tail
0x21e8-0x21eb	Receive Selector List 15 Count
0x21ec-0x21ef	Reserved
0x21f0-0x21f3	Receive Selector List 16 Head
0x21f4-0x21f7	Receive Selector List 16 Tail
0x21f8-0x21fb	Receive Selector List 16 Count
0x21fc-0x21ff	Reserved
0x2200-0x2203	Local Statistics Counter: Class of Service 1

Table 324: Receive List Placement Control Registers (Cont.)

Offset	***Registers***
0x2204-0x2207	Local Statistics Counter: Class of Service 2
0x2208-0x220b	Local Statistics Counter: Class of Service 3
0x220c-0x220f	Local Statistics Counter: Class of Service 4
0x2210-0x2213	Local Statistics Counter: Class of Service 5
0x2214-0x2217	Local Statistics Counter: Class of Service 6
0x2218-0x221b	Local Statistics Counter: Class of Service 7
0x221c-0x221f	Local Statistics Counter: Class of Service 8
0x2220-0x2223	Local Statistics Counter: Class of Service 9
0x2224-0x2227	Local Statistics Counter: Class of Service 10
0x2228-0x222b	Local Statistics Counter: Class of Service 11
0x222c-0x222f	Local Statistics Counter: Class of Service 12
0x2230-0x2233	Local Statistics Counter: Class of Service 13
0x2234-0x2237	Local Statistics Counter: Class of Service 14
0x2238-0x223b	Local Statistics Counter: Class of Service 15
0x223c-0x223f	Local Statistics Counter: Class of Service 16
0x2240-0x2243	Local Statistics Counter: Drop Due to Filter
0x2244-0x2247	Local Statistics Counter: DMA Write Queue Full
0x2248-0x224b	Local Statistics Counter: DMA High Priority Write Queue Full
0x224c-0x224f	Local Statistics Counter: No More Receive BD
0x2250-0x2253	Local Statistics Counter: ifInDiscards
0x2254-0x2257	Local Statistics Counter: ifInErrors
0x2258-0x225b	Local Statistics Counter: Receive Threshold Hit
0x225c-0x23ff	Reserved

Receive List Placement Mode Register (Offset 0x2000)

Table 325: Receive List Placement Mode Register (Offset 0x2000)

Bit	Field	Description	Init	Access
31-5	Reserved	-	0	R/O
4	Stats Overflow Attn Enable	Enable attention for statistics overflow.	-	R/W
3	Mapping Out of Range Attn Enable	Enable attention for mapping out of range error.	-	R/W
2	Class Zero Attn Enable	Enable attention for zero class field.	-	R/W
1	Enable	This bit controls whether the Receive List Placement state machine is active or not. When set to 0, it completes the current operation and cleanly halts. Until it is completely halted, it remains one when read.	1	R/W
0	Reset	When this bit is set to 1, the Receive List Placement state machine is reset. This is a self-clearing bit.	0	R/W

Receive List Placement Status Register (Offset 0x2004)

Table 326: Receive List Placement Status Register (Offset 0x2004)

Bit	Field	Description	Init	Access
31-5	Reserved	-	0	R/O
4	Stats Overflow Attn	A statistics managed by Receive List Placement has overflowed.	-	R/O
3	Mapping Out of Range Attn	Class of service mapping is out of the range of the active queue number.	-	R/O
2	Class Zero Attn	Class field extracted from frame descriptor is zero.	-	R/O
1-0	Reserved	-	0	R/O

RECEIVE SELECTOR LIST LOCK REGISTER (OFFSET 0X2008)

This 32-bit register is used by the RISCs to obtain exclusive access to a selector list head, tail, and counter. Bits 0 and 16 refer to receive selector list 1. Bits 31 and 15 refer to receive selector list 16. To use this register, set the appropriate request bit and then read back the register checking the associated grant bit. If the grant bit is set, the lock has been obtained. To free the lock, reset the request bit. If the request bit is 1 but the grant bit is 0, then a request to lock that particular list is pending. Similarly, if the request bit is 0 but the grant bit is 1, then a request to free that particular list is pending. A request to lock a particular list is successful when both request and grant bits are 1. Similarly, a request to free a particular list is successful when both request and grant bits are 0.

Table 327: Receive Selector List Lock Register (Offset 0x2008)

Bit	Field	Description	Init	Access
31-16	Grant Bits	Each bit is mapped to indicate that a CPU currently has locked a particular selector list's head, tail, and count register.	-	R/O
15-0	Request bits	Each bit is mapped to allow a CPU to request a lock for a particular selector list's head, tail, and count register. When a request bit is set, the hardware attempts to obtain the associated lock. When successful, the associated grant bit is set. When unsuccessful, the associated grant bit is not set.	-	R/W

RECEIVE SELECTOR NON-EMPTY BITS REGISTER (OFFSET 0X200C)

This 32-bit register is used by the RISCs to quickly determine the status of the receive selector. Bit 0 refers to receive selector list 1. Bit 15 refers to receive selector list 16. If this register is nonzero the receive selector non-empty bit is set in the RX-CPU event register.

Table 328: Receive Selector Non-Empty Bits Register (Offset 0x200C)

Bit	Field	Description	Init	Access
31-16	Reserved	-	0	R/O
15-0	List non-empty bits	If set, the bit indicates that the associated list is not empty (that is the counter is nonzero).		R/O

Receive List Placement Configuration Register (Offset 0x2010)

Table 329: Receive List Placement Configuration Register (Offset 0x2010)

Bit	Field	Description	Init	Access
31-15	Reserved	-	0	R/O
14-13	Default Interrupt Distribution Queue	Default interrupt distribution queue. Number within a class of service group when the frame has errors, is truncated, or is a non-IP frame.	00	R/W
12-8	Bad Frames Class (BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	Default class for error or truncated frames. These frames are placed in this class of service group when the Allow Bad Frame bit (bit 11) is set in the Mode Control register.	00001	R/O
	Bad Frames Class (other devices)	Default class for error or truncated frames. These frames are placed in this class of service group when the Allow Bad Frame bit (bit 11) is set in the Mode Control register.	00000	R/W
7-3	Number of Active Lists	The total number of active receive lists. The value must be between 1 and 16. This value must be an integer multiple of the Number of Lists per Distribution Group value.	00000	R/W
2-0	Number of Lists Per Distribution Group	Specifies the number of lists per interrupt distribution group. This register must always be a power of 2. For example, if the system wants four classes of service and four interrupt distribution lists per class of service, this value is set to four and the Number of Active Lists value is set to 16.	000	R/W

RECEIVE LIST PLACEMENT STATISTICS CONTROL REGISTER (OFFSET 0X2014)

Table 330: Receive List Placement Statistics Control Register (Offset 0x2014)

Bit	Field	Description	Init	Access
31-5	Reserved	-	0	R/O
4	Reserved (BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	-	0	R/O
	Force Statistics Zero (other devices)	If set, forces the statistics in NIC memory to zeros. This should be done when statistics enable is low. Only the masked statistics will be cleared. This is a self-clearing bit.	0	R/W
3	Reserved (BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	-	0	R/O
	Force Statistics Flush (other devices)	When set, forces a flush of the local statistics to the NIC memory by doing a read-modify-write operation. This can be set only when statistics enable is low. Flushes only masked statistics. Self-clearing when flush completes.	0	R/W
2	Statistics Clear	When set, resets local statistics counters to zero. Clears only masked statistics. Self-clearing when done.	0	R/W
1	Reserved (BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	-	0	R/O
	Faster Statistics Update (other devices)	Allows a faster update of the statistics counters to the NIC memory. When set, one statistics is updated every 15 us (or 998 clocks). When not set, one statistics is updated every 25 us (or 1662 clocks).		R/W
0	Statistics Enable	When set, allow the local statistics counters to increment. When reset, counters hold their values until the next update to the NIC memory. Enables only masked statistics.	0	R/W

Receive List Placement Statistics Enable Mask Register (Offset 0x2018)

This version of the Receive List Placement Statistics Enable Mask register applies to the BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 devices only.

Table 331: Receive List Placement Statistics Enable Mask Register (Offset 0x2018)

Bit	***Field***	***Description***	***Init***	***Access***
31-23	Reserved	–	0	R/O
22	Dynamic Switching LSO During Long Burst Read Fix (BCM5751 and BCM5721 A2 versions only).	Dynamic switching LSO during long burst read bug fix enable. Set to 0 to enable the fix. This bit is valid for A2 only.		R/W
21-19	Reserved	–	0	R/O
18	Static Switching LSO During Long Burst Read Fix (BCM5721 and BCM5751 A2 only)	Static switching LSO during long burst read bug fix enable. Set to 0 to enable the fix. This bit is valid for A2 only.	0	R/W
	Disable MACTQ Double Ack Issue Fix (BCM5752, BCM5714, BCM5715, B1 and later versions of BCM5751, and B1 and later versions of BCM5721 devices only)	Disable MACTQ double Ack issue fix. • 1: Disabled • 0: Enabled	1	R/W
	Reserved (Other devices)	–	0	R/O
17-2		BCM57XX ASIC Revision ID. The value is: • 0x3001 for A1 • 0x0000 for A0 ***Note:*** See "Revision Levels" on page 5.		R/W
1	Keep CLKRUN Behavior the Same as A0 (BCM5721 and BCM5751 A1 version only)	When this bit is set, the chip behaves the same as in A0. $\overline{\text{CLKRUN}}$ is not expected to work unless Clock Control register (see "PCI Clock Control Register (Offset 0x74)" on page 334) bit 21 is set. This bit is valid for A1 only.	1	R/W
	Disable ASF Lockup Issue Fix (B1 and later versions of BCM5751 and BCM5721 only)	Disable ASF lockup fix. • 1: Disabled • 0: Enabled	1	R/W
0	Reserved	–	0	R/O

Rest of BCM57XX Family

This version of the Receive List Placement Statistics Enable Mask register applies to the rest of the BCM57XX family.

Table 332: Receive List Placement Stat. Enable Mask (Offset 0x2018, Rest of BCM57XX Family)

Bit	Field	Description	Init	Access
31-23	Reserved	–	0	R/O
22-0	Counters Enable Mask	Controls which statistics can be updated, cleared, or flushed. • Bit 0-15 corresponds to statistics for Class of Service 1-16. • Bit 16-22 correspond to statistics for Drop due to filter, DMA Write Queue Full, DMA High Priority Write Queue Full, No More Receive BD, ifInDiscards, ifInErrors, and Receive Threshold Hit. • 1 = Inaccessible • 0 = Accessible		R/W

Receive List Placement Statistics Increment Mask Register (Offset 0x201C)

This version of the Receive List Placement Statistics Increment Mask register applies to the BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 devices only.

Table 333: Receive List Placement Statistics Increment Mask Register (Offset 0x201C)

Bit	Field	Description	Init	Access
31-22	Reserved	-	0	R/O
21-16	Counters Increment Mask	Writing a 1 to a Counters Increment Mask bit forces the corresponding statistics counter to increment by 1. Not affected by Statistics Enable Mask. Bits 16-21 correspond to statistics for Drop due to filter, DMA Write Queue Full, DMA High Priority Write Queue Full, No More Receive BD, ifInDiscards, and ifInErrors.	0	W/O
15-1	Reserved	–	0	R/O
0	Counters Increment Mask	Writing a 1 to a Counters Increment Mask bit forces the corresponding statistics counter to increment by 1. Not affected by Statistics Enable Mask. Bit 0 corresponds to statistics Class of Service 1.	0	W/O

Rest of BCM57XX Family

This version of the Receive List Placement Statistics Increment Mask register applies to the rest of the BCM57XX family.

Table 334: Receive List Placement Stat. Increment Mask (Offset 0x201C, Rest of BCM57XX Fam.)

Bit	Field	Description	Init	Access
31-23	Reserved	-	0	R/O
22-0	Counters Increment Mask	Writes a 1, to bit position forces the corresponding statistics counter to increment by 1. Not affected by Statistics Enable Mask. Bits 0-15 correspond to statistics for Class of Service 1-16. Bits 16-22 correspond to statistics for Drop due to filter, DMA Write Queue Full, DMA High Priority Write Queue Full, No More Receive BD, ifInDiscards, ifInErrors, and Receive Threshold Hit.	0	W/O

RECEIVE SELECTOR LIST HEAD AND TAIL POINTERS (OFFSETS STARTING AT 0X2100)

The 16 receive selector lists head and tail pointers are MBUF cluster pointers. The selector list head pointer is the MBUF cluster pointer of the first frame queued in the associated selector list. Similarly, the selector list tail pointer is the MBUF cluster pointer of the last frame queued in that selector list.

RECEIVE SELECTOR LIST COUNT REGISTERS (OFFSET OF LIST N: 0X2108 + 16*[N-1])

These registers indicate how many frames are currently queued to the associated selector list.

LOCAL STATISTICS COUNTER REGISTER (OFFSET 0X2200-0X2258)

Table 335: Local Statistics Counter (Offset 0x2200)

Bit	*Field*	*Description*	*Init*	*Access*
31-10	Reserved	-	0	R/O
9-0	Counters Value	The current counter value for statistics kept by the Receive List Placement.		R/O

RECEIVE DATA AND RECEIVE BD INITIATOR CONTROL REGISTERS

Table 336: Receive Data and Receive BD Initiator Control Registers

Offset	*Registers*
0x2400-0x2403	Receive Data and Receive BD Ring Initiator Mode
0x2404-0x2407	Receive Data and Receive BD Ring Initiator Status
0x2408-0x240b	Reserved
0x240c-0x240f	Reserved
0x2440-0x244f	Jumbo Receive BD ring RCB
0x2450-0x245f	Standard Receive BD ring RCB
0x2460-0x246f	Mini Receive BD ring RCB
0x2470-0x2473	Receive Diagnostic Data and Receive BD Ring Initiator Local NIC Jumbo Receive BD Consumer Index
0x2474-0x2477	Receive Diagnostic Data and Receive BD Ring Initiator Local NIC Standard Receive BD Consumer Index
0x2478-0x247b	Receive Diagnostic Data and Receive BD Ring Initiator Local NIC Mini Receive BD Consumer Index
0x247c-0x247f	Reserved
0x2480-0x2483	Receive Diagnostic Data and Receive BD Initiator Local Receive Return 1 Producer Index
0x2484-0x2487	Receive Diagnostic Data and Receive BD Initiator Local Receive Return 2 Producer Index
0x2488-0x248b	Receive Diagnostic Data and Receive BD Initiator Local Receive Return 3 Producer Index
0x248c-0x248f	Receive Diagnostic Data and Receive BD Initiator Local Receive Return 4 Producer Index
0x2490-0x2493	Receive Diagnostic Data and Receive BD Initiator Local Receive Return 5 Producer Index
0x2494-0x2497	Receive Diagnostic Data and Receive BD Initiator Local Receive Return 6 Producer Index
0x2498-0x249b	Receive Diagnostic Data and Receive BD Initiator Local Receive Return 7 Producer Index
0x249c-0x249f	Receive Diagnostic Data and Receive BD Initiator Local Receive Return 8 Producer Index
0x24a0-0x24a3	Receive Diagnostic Data and Receive BD Initiator Local Receive Return 9 Producer Index
0x24a4-0x24a7	Receive Diagnostic Data and Receive BD Initiator Local Receive Return 10 Producer Index
0x24a8-0x24ab	Receive Diagnostic Data and Receive BD Initiator Local Receive Return 11 Producer Index
0x24ac-0x24af	Receive Diagnostic Data and Receive BD Initiator Local Receive Return 12 Producer Index
0x24b0-0x24b3	Receive Diagnostic Data and Receive BD Initiator Local Receive Return 13 Producer Index
0x24b4-0x24b7	Receive Diagnostic Data and Receive BD Initiator Local Receive Return 14 Producer Index
0x24b8-0x24bb	Receive Diagnostic Data and Receive BD Initiator Local Receive Return 15 Producer Index
0x24bc-0x24bf	Receive Diagnostic Data and Receive BD Initiator Local Receive Return 16 Producer Index
0x24c0-0x24c3	Receive Diagnostic Data and Receive BD Initiator Hardware Diagnostic
0x24c4-0x27ff	Reserved

RECEIVE DATA AND RECEIVE BD INITIATOR MODE REGISTER (OFFSET 0X2400)

Table 337: Receive Data and Receive BD Initiator Mode Register (Offset 0x2400)

Bit	*Field*	*Description*	*Init*	*Access*
31-8	Reserved	-	0	R/O
7	RDI Timer Event Enable (BCM5714 and BCM5715 only)	Enables the RDI timer attention.	0	R/W
6-5	Reserved	-	0	R/O
4	Illegal return ring size	Enables illegal return ring size attention.		R/W
3	Frame size is too large to fit into one Receive BD	Enables frame size is too large to fit into one Receive BD attention.		R/W
2	Reserved (BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	-	0	R/O
	Jumbo Receive BD is needed and Jumbo Receive BD ring is disabled (other devices)	Enables Jumbo Receive BD is needed and Jumbo Receive BD ring is disabled attention.		R/W
1	Enable	This bit controls whether the Receive Data and Receive BD Initiator state machine is active or not. When set to 0, it completes the current operation and cleanly halts. Until it is completely halted, it remains one when read.		R/W
0	Reset	When this bit is set to 1, the Receive Data and Receive BD Initiator state machine is reset. This is a self-clearing bit.		R/W

RECEIVE DATA AND RECEIVE BD INITIATOR STATUS REGISTER (OFFSET 0X2404)

Table 338: Receive Data and Receive BD Initiator Status Register (Offset 0x2404)

Bit	*Field*	*Description*	*Init*	*Access*
31-8	Reserved	-	0	R/O
7	RDI Timer Attention (BCM5714 and BCM5715 only)	This bit is asserted if the RDI module does not process an RDI-FTQ entry within the time specified in the RDI Timer Mode Register (0x24F0). This attention indicates that the Receive data path is stalled.	0	R/W
6-5	Reserved	-	0	R/O
4	Illegal return ring size	One of the return rings contains illegal ring size (e.g., only contains 1024 entries)		R/O
3	Frame size is too large to fit into one Receive BD	The received frame size is too big for the selected Receive BD.		R/O

Table 338: Receive Data and Receive BD Initiator Status Register (Offset 0x2404) (Cont.)

Bit	Field	Description	Init	Access
2	Reserved (BCM5705 BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	-	0	R/O
	Jumbo Receive BD is needed and Jumbo Receive BD ring is disabled (other devices)	The received frame's size exceeds the capacity of the standard Receive BD, and Jumbo Receive BD ring is disabled.		R/O
1-0	Reserved	-		

JUMBO RECEIVE BD RING RCB REGISTER (OFFSET 0X2440)

These registers are not applicable to BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 devices.

Table 339: Receive Producer Ring Host Address High Register (Offset 0x2440)

Bit	Field	Description	Init	Access
31-0	Host Address High	The host ring address is the host address of the first ring element. The host ring address is in host address format.	00000000h	R/W

Table 340: Receive Producer Ring Host Address Low Register (Offset 0x2444)

Bit	Field	Description	Init	Access
31-0	Host Address Low	The host ring address is the host address of the first ring element. The host ring address is in host address format.	00000000h	R/W

Table 341: Receive Producer Length/Flags Register (Offset 0x2448)

Bit	Field	Description	Init	Access
31-16	Max Length	Unused for jumbo rings; otherwise, specifies the maximum size of an Ethernet packet plus VLAN tag.	00000000h	R/W
15-2	Reserved	Unused.	000h	R/W
1	Disable Ring	Set to disable the use of the ring.	0	R/W
0	Extended RX Enable	Set to use the extended receive buffer descriptors.	0	R/W

Table 342: Receive Producer Ring NIC Address (Offset 0x244C)

Bit	Field	Description	Init	Access
31-0	NIC Address	The NIC ring address is the NIC address of the first ring element.	00000000h	R/W

STANDARD RECEIVE BD RING RCB REGISTER (OFFSET 0X2450)

Same as above Jumbo ring RCB with address offset from 0x2450–0x245fh.

MINI RECEIVE BD RING RCB REGISTER (OFFSET 0X2460)

Same as above Jumbo ring RCB with address offset from 0x2460–0x246fh. These registers are only applicable to BCM5700 device.

RECEIVE DIAGNOSTIC DATA AND RECEIVE BD RING INITIATOR LOCAL NIC JUMBO RECEIVE BD CONSUMER INDEX (OFFSET 0X2470)

This set of registers keeps track of the current DMAs queued to move receive data from the NIC to the host.

This register is not applicable to BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 devices.

The Receive Data and Receive BD Initiator maintains the state of the indices by keeping two local copies, a copy of the NIC's return ring producer index, and a copy of the NIC's receive BD consumer index. The local return ring producer index is set to the value placed in the DMA descriptor. The local NIC receive return consumer index is also set to the value placed in the DMA descriptor.

There is a local copy of each of the three receive BD producer indices. There is also a local copy of each of 16 NIC receive return consumer indices.

RECEIVE DIAGNOSTIC DATA AND RECEIVE BD RING INITIATOR LOCAL NIC STANDARD RECEIVE BD CONSUMER INDEX (OFFSET 0X2474)

Same as "Receive Diagnostic Data and Receive BD Ring Initiator Local NIC Jumbo Receive BD Consumer Index (Offset 0x2470)" on page 439.

RECEIVE DIAGNOSTIC DATA AND RECEIVE BD RING INITIATOR LOCAL NIC MINI RECEIVE BD CONSUMER INDEX (OFFSET 0X2478)

Same as "Receive Diagnostic Data and Receive BD Ring Initiator Local NIC Jumbo Receive BD Consumer Index (Offset 0x2470)" on page 439.This register is applicable to BCM5700 device only.

RECEIVE DATA AND RECEIVE DIAGNOSTIC BD INITIATOR LOCAL RECEIVE RETURN PRODUCER INDEX REGISTER (OFFSET 0X2480-0X24BC)

Same as "Receive Diagnostic Data and Receive BD Ring Initiator Local NIC Jumbo Receive BD Consumer Index (Offset 0x2470)" on page 439. The registers from 0x2484 to 0x24BC are not applicable to BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 devices.

RECEIVE DATA AND RECEIVE BD INITIATOR HARDWARE DIAGNOSTIC REGISTER (OFFSET 0X24C0)

Table 343: Receive Data and Receive BD Initiator Hardware Diagnostic Register (Offset 0x24C0)

Bit	Field	Description	Init	Access
31-0	Diagnostics	Hardware Diagnostics.	0	R/O

RDI TIMER MODE REGISTER (0X024F0H)

Table 344: RDI Timer Mode Register (0x024F0h, BCM5714 and BCM5715 Only)

Bit	Field	Description	Init	Access
0	Rdi_timer_ctl_reset	Resets only the Rdi_timer control module.	0	R/W
1	Rdi_timer_ctl_en	RDI timer module enable.	0	R/W
2	Rdi_timer_cnt reset	Reset only the rdi_timer (debug only).	0	R/W
3	Reserved	-	0	R
5:4	RDI Attn time out value	The time an RDI FTQ entry is not processed before generating Rdi_timer_attn. • 00 = 1s • 01 = 2s • 10 = 3s • 11 = Reserved	00	R/W
7:4	RDI timer clk sel	Debug purpose only. • 00 = 640-ns clock • 01 = 40-ns clock • 10 = 10240-ns clock • 11 = Core clock	00	R/W

RECEIVE DATA COMPLETION CONTROL REGISTERS

Table 345: Receive Data Completion Control Registers

Offset	Registers
0x2800-0x2803	Receive Data Completion Mode
0x2804-0x2bff	Reserved

RECEIVE DATA COMPLETION MODE REGISTER (OFFSET 0X2800)

Table 346: Receive Data Completion Mode Register (Offset 0x2800)

Bit	Field	Description	Init	Access
31-3	Reserved	-	0	R/O
2	Attn_Enable	When this bit is set to 1, an internal attention is generated when an error occurs.	0	R/W
1	Enable	This bit controls whether the Receive Data Completion state machine is active or not. When set to 0, it completes the current operation and cleanly halts. Until it is completely halted, it remains one when read.	1	R/W
0	Reset	When this bit is set to 1, the Receive Data Completion state machine is reset. This is a self-clearing bit.	0	R/W

RECEIVE BD INITIATOR CONTROL REGISTERS

Table 347: Receive BD Initiator Control Registers

Offset	Registers
0x2c00-0x2c03	Receive BD Initiator Mode
0x2c04-0x2c07	Receive BD Initiator Status
0x2c08-0x2c0b	Receive BD Initiator Local NIC Jumbo Receive BD Producer Index
0x2c0c-0x2c0f	Receive BD Initiator Local NIC Standard Receive BD Producer Index
0x2c10-0x2c13	Receive BD Initiator Local NIC Mini Receive BD Producer Index
0x2c14-0x2c17	Mini Receive BD Ring Replenish Threshold
0x2c18-0x2c1b	Standard Receive BD Ring Replenish Threshold
0x2c1c-0x2c1f	Jumbo Receive BD Ring Replenish Threshold
0x2c20-0x2fff	Reserved

RECEIVE BD INITIATOR MODE REGISTER (OFFSET 0X2C00)

Table 348: Receive Data Initiator Mode Register (Offset 0x2C00)

Bit	Field	Description	Init	Access
31-3	Reserved	-	0	R/O
2	Receive BDs available on a disabled Receive BD ring enable	Attention enable for Receive BDs available on a disabled Receive BD ring.		R/W
1	Enable	This bit controls whether the Receive BD Initiator state machine is active or not. When set to 0, it completes the current operation and cleanly halts. Until it is completely halted, it remains one when read.	1	R/W
0	Reset	When this bit is set to 1, the Receive BD Initiator state machine is reset. This is a self-clearing bit.	0	R/W

RECEIVE BD INITIATOR STATUS REGISTER (OFFSET 0X2C04)

Table 349: Receive BD Initiator Status Register (Offset 0x2C04)

Bit	Field	Description	Init	Access
31-3	Reserved	-	0	R/O
2	Receive BDs available on a disabled Receive BD ring status	Host requests to DMA Receive BDs to a disabled ring.		R/O
1-0	Reserved	-	0	R/O

RECEIVE BD INITIATOR LOCAL NIC RECEIVE BD PRODUCER INDEX REGISTERS (OFFSET 0X2C08-0X2C13)

This set of registers is used to keep track of the current DMAs queued to move receive BDs from the host to the NIC.

MINI RECEIVE BD PRODUCER RING REPLENISH THRESHOLD REGISTER (OFFSET 0X2C14)

This set of registers is used to keep the Receive Initiator BD state machine from generating a large number of DMA requests for receive buffer descriptors. Each indicates the number of buffer descriptors that must be indicated before a DMA is initiated. This register is applicable to BCM5700 device only.

Table 350: Mini Receive BD Producer Ring Replenish Threshold Register (Offset 0x2C14)

Bit	Field	Description	Init	Access
31-0	BD Num	Number of buffer descriptors indicated by the receive producer index for the DMA engine to initiate a transfer of buffer descriptors for replenishing the ring.	00000000h	R/W

STANDARD RECEIVE BD PRODUCER RING REPLENISH THRESHOLD REGISTER (OFFSET 0X2C18)

Table 351: Standard Receive BD Producer Ring Replenish Threshold Register (Offset 0x2C18)

Bit	Field	Description	Init	Access
31-0	BD Num	Number of buffer descriptors indicated by the receive producer index for the DMA engine to initiate a transfer of buffer descriptors for replenishing the ring.	00000000h	R/W

JUMBO RECEIVE BD PRODUCER RING REPLENISH THRESHOLD REGISTER (OFFSET 0X2C1C)

This register is not applicable to BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 devices.

Table 352: Jumbo Receive BD Producer Ring Replenish Threshold Register (Offset 0x2C1C)

Bit	Field	Description	Init	Access
31-0	BD Num	Number of buffer descriptors indicated by the receive producer index for the DMA engine to initiate a transfer of buffer descriptors for replenishing the ring.	00000000h	R/W

RECEIVE BD COMPLETION CONTROL REGISTERS

Table 353: Receive BD Completion Control Registers

Offset	Registers
0x3000-0x3003	Receive BD Completion Mode
0x3004-0x3007	Receive BD Completion Status
0x3008-0x300b	NIC Jumbo Receive BD Producer Index
0x300c-0x300f	NIC Standard Receive BD Producer Index
0x3010-0x3013	NIC Mini Receive BD Producer Index
0x3014-0x33ff	Reserved

RECEIVE BD COMPLETION MODE REGISTER (OFFSET 0X3000)

Table 354: Receive BD Completion Mode Register (Offset 0x3000)

Bit	Field	Description	Init	Access
31-3	Reserved	-	0	R/O
2	Attn_Enable	When this bit is set to 1, an internal attention is generated when an error occurs.		R/W
1	Enable	This bit controls whether the Receive BD Completion state machine is active or not. When set to 0, it completes the current operation and cleanly halts. Until it is completely halted, it remains one when read.	1	R/W
0	Reset	When this bit is set to 1, the Receive BD Completion state machine is reset. This is a self-clearing bit.	0	R/W

RECEIVE BD COMPLETION STATUS REGISTER (OFFSET 0X3004)

Table 355: Receive BD Completion Status Register (Offset 0x3004)

Bit	Field	Description	Init	Access
31-3	Reserved	-	0	R/O
2	Error	Receive BD Completion error status.		R/O
1-0	Reserved	-	0	R/O

NIC Jumbo Receive BD Producer Index Register (Offset 0x3008)

This register is not applicable to BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 devices.

Table 356: NIC Jumbo Receive BD Producer Index (Offset 0x3008)

Bit	Field	Description	Init	Access
31-8	Reserved	-	0	R/O
7-0	NIC Jumbo Receive BD Producer Index	-		R/W

NIC Standard Receive BD Producer Index Register (Offset 0x300C)

Table 357: NIC Standard Receive BD Producer Index (Offset 0x300C)

Bit	Field	Description	Init	Access
31-9	Reserved	-	0	R/O
8-0	NIC Standard Receive BD Producer Index	-	-	R/W

NIC Mini Receive BD Producer Index Register (Offset 0x3010)

This register is applicable to BCM5700 device only.

Table 358: NIC Mini Receive BD Producer Index (Offset 0x3010)

Bit	Field	Description	Init	Access
31-10	Reserved	-	0	R/O
9-0	NIC Mini Receive BD Producer Index	-	-	R/W

RECEIVE LIST SELECTOR CONTROL REGISTERS

Table 359: Receive List Selector Control Registers

Offset	Registers
0x3400-0x3403	Receive List Selector Mode.
0x3404-0x3407	Receive List Selector Status.
0x3408-0x37ff	Reserved.

RECEIVE LIST SELECTOR MODE REGISTER (OFFSET 0X3400)

Table 360: Receive List Selector Mode Register (Offset 0x3400)

Bit	Field	Description	Init	Access
31-3	Reserved	-	0	R/O
2	Attn_Enable	When this bit is set to 1, an internal attention is generated when an error occurs.		R/W
1	Enable	This bit controls whether the Receive List Selector state machine is active or not. When set to 0, it completes the current operation and cleanly halts. Until it is completely halted, it remains one when read.	1	R/W
0	Reset	When this bit is set to 1, the Receive List Selector state machine is reset. This is a self-clearing bit.	0	R/W

RECEIVE LIST SELECTOR STATUS REGISTER (OFFSET 0X3404)

Table 361: Receive List Selector Status Register (Offset 0x3404)

Bit	Field	Description	Init	Access
31-3	Reserved	-	0	R/O
2	Error	Receive List Selector error status.		R/O
1-0	Reserved	-	0	R/O

MBUF CLUSTER FREE REGISTERS

These registers are applicable to BCM5700, BCM5701, BCM5702, BCM5703C, BCM5703S, BCM5704C, and BCM5704S devices only.

Table 362: MBUF Cluster Free Registers

Offset	*Registers*
0x3800-0x3803	MBUF Cluster Free Mode.
0x3804-0x3807	MBUF Cluster Free Status.
0x3808-0x3bff	Reserved.

MBUF CLUSTER FREE MODE REGISTER (OFFSET 0X3800)

Table 363: MBUF Cluster Free Mode Register (Offset 0x3800)

Bit	*Field*	*Description*	*Init*	*Access*
31-3	Reserved	-	0	R/O
2	Attn_Enable	When this bit is set to 1, an internal attention is generated when an error occurs.		R/W
1	Enable	This bit controls whether the MBUF Cluster Free state machine is active or not. When set to 0, it completes the current operation and cleanly halts. Until it is completely halted, it remains one when read.	1	R/W
0	Reset	When this bit is set to 1, the MBUF Cluster Free state machine is reset. This is a self-clearing bit.	0	R/W

MBUF CLUSTER FREE STATUS REGISTER (OFFSET 0X3804)

Table 364: MBUF Cluster Free Status Register (Offset 0x3804)

Bit	*Field*	*Description*	*Init*	*Access*
31-3	Reserved	-	0	R/O
2	Error	MBUF Cluster error status.		R/O
1-0	Reserved	-	0	R/O

DBU REGISTERS

These DBU registers are applicable only to the BCM5752.

Table 365: DBU Command Register (0x3800)

Bit	Field	Description	Init	Access
2	RX overflow	RX overflow error. Remains set until a 1 is written.	0	W2C
1	RX error	RX framing error. Remains set until a 1 is written.	0	W2C
0	Enabled	Always reads 1 because the DBU cannot be disabled.	1	RO

Table 366: DBU Status Register (0x3804)

Field	Bit	Access	Init	Description
TX data occupied	1	RO	0	This bit is set automatically when the TX data register is written, and remains set until the written character is transmitted, at which point it clears automatically. Firmware can poll this bit to determine when it is acceptable to write the next TX character.
RX data valid	0	RO	0	This bit is set automatically when a character is received, and remains set until the received character is read from the RX Data register, at which point it clears automatically. Firmware can poll this bit to determine when there is a received character to be retrieved.

Table 367: DBU Configuration Register (0x3808)

Bit	Field	Description	Init	Access
4	DBU MA bypass	When this bit is set, the CDARB block is completely bypassed, causing the DBU to hang when trying to access memory space. This feature is included as a workaround in the event that there is a bug in CDARB. Setting it should allow CP access to MA without interference.	0	RW
3	DBU MA transparent	This bit places the CDARB in transparent mode. When set, the CDARB allows a read to start while a write is still in progress, or a write to start when a read is still in progress. If the bit is clear, then the CDARB waits until the current MA transaction is complete before starting a new transaction in a different direction.	1	RW
2	CRLF enable	When this bit is set, all transmitted line feeds are preceded by a carriage return. When it is clear, only the line feed is transmitted.	1	RW
1	Debug state machine enable	When this bit is set, the state machine debugger is enabled and responds to commands typed by echoing characters, initiating GRC or memory read/write cycles, and responding with returned data. When this bit is clear, the debugger is disabled. This bit should only be cleared if firmware is running to respond to receive and transmit characters. **IMPORTANT!** If this bit is cleared via the debugger, there is no way to set it again via the debugger.	1	RW
0	Timing override	If this bit is cleared, then the baud rate is 19200 (assuming CK25 is 25 MHz). If the bit is set, then baud rate timing is set based on the values in the timing register.	0	RW

Table 368: DBU Timing Register (0x380C)

Bit	Field	Description	Init	Access
31:16	Bit interval	This field sets the number of CK25 cycles between serial bits for both transmit and receive data. The default value results in a baud rate of 19200 for a 25-MHz CK25 clock. ***Note:*** This value has no effect unless the timing override bit is set in the configuration register.	0x516	RW
15:0	First bit sample offset	This field sets the number of CK25 cycles from the falling edge of serial in to the point where the start bit is sampled. The default value results in a baud rate of 19200 for a 25-MHz CK25 clock. ***Note:*** This value has no effect unless the timing override bit is set in the configuration register.	0x28B	RW

Table 369: DBU RX Data Register (0x3810)

Bit	Field	Description	Init	Access
8	RX data error	This bit indicates that the data in bits 7:0 was received with a framing error. The value in this bit is valid only when the RX Data Valid bit is also set in Table 366 on page 448. The act of reading this register automatically clears the RX Data Valid bit in the status register.	N/A	RO
7:0	RX data	These bits contain the last received serial character. The value in these bits is only valid when the RX Data Valid bit is also set in Table 366 on page 448. The act of reading this register automatically clears the RX Data Valid bit in the status register.	N/A	RO

Table 370: DBU TX Data Register (0x3814)

Bit	Field	Description	Init	Access
7:0	TX data	These bits can be written with a character to be transmitted. When this register is written, transmission of a serial character commences. The act of writing this register automatically sets the txdata_occupied bit in the status register. The occupied bit remains set until the UART is ready to accept another character to transmit. Firmware should always check the state of txdata_occupied before writing this register; otherwise, characters that have not yet been transmitted may be overwritten. This register reads back the last character that was written.	N/A	R/W

HOST COALESCING CONTROL REGISTERS

The Host Coalescing Control Registers are responsible for pacing the rate at which the NIC updates the host's transmit and receive buffer descriptor ring indices. Although the host produces and receives frames in one or more buffer descriptors, the Host Coalescing state machine always updates the host on frame boundaries. Additionally, the Host Coalescing state machine regulates the rate at which the statistics are updated in host memory.

Table 371: Host Coalescing Control Registers

Offset	*Registers*
0x3c00-0x3c03	Host Coalescing Mode
0x3c04-0x3c07	Host Coalescing Status
0x3c08-0x3c0b	Receive Coalescing Ticks
0x3c0c-0x3c0f	Send Coalescing Ticks
0x3c10-0x3c13	Receive Max Coalesced BD Count
0x3c14-0x3c17	Send Max Coalesced BD Count
0x3c18-0x3c1b	Receive Coalescing Ticks, during interrupt
0x3c1b-0x3c1f	Send Coalescing Ticks, during interrupt
0x3c20-0x3c23	Receive Max Coalesced BD Count, during interrupt
0x3c24-0x3c27	Send Max Coalesced BD Count, during interrupt
0x3c28-0x3c2b	Statistics Ticks
0x3c2c-0x3c2f	Reserved
0x3c30-0x3c37	Statistics Host Address
0x3c38-0x3c3f	Status Block Host Address
0x3c40-0x3c43	Statistics Base Address
0x3c44-0x3c47	Status Block Base Address
0x3c48-0x3c4b	Flow Attention Register
0x3c4c-0x3c4f	Reserved
0x3c50-0x3c53	NIC Jumbo Receive BD Consumer Index
0x3c54-0x3c57	NIC Standard Receive BD Consumer Index
0x3c58-0x3c5b	NIC Mini Receive BD Consumer Index
0x3c5c-0x3c7f	Reserved
0x3c80-0x3c83	NIC Diagnostic Return Ring Producer Index 1
0x3c84-0x3c87	NIC Diagnostic Return Ring Producer Index 2
0x3c88-0x3c8b	NIC Diagnostic Return Ring Producer Index 3
0x3c8c-0x3c8f	NIC Diagnostic Return Ring Producer Index 4
0x3c90-0x3c93	NIC Diagnostic Return Ring Producer Index 5
0x3c94-0x3c97	NIC Diagnostic Return Ring Producer Index 6
0x3c98-0x3c9b	NIC Diagnostic Return Ring Producer Index 7
0x3c9c-0x3c9f	NIC Diagnostic Return Ring Producer Index 8
0x3ca0-0x3ca3	NIC Diagnostic Return Ring Producer Index 9

Table 371: Host Coalescing Control Registers (Cont.)

Offset	Registers
0x3ca4-0x3ca7	NIC Diagnostic Return Ring Producer Index 10
0x3ca8-0x3cab	NIC Diagnostic Return Ring Producer Index 11
0x3cac-0x3caf	NIC Diagnostic Return Ring Producer Index 12
0x3cb0-0x3cb3	NIC Diagnostic Return Ring Producer Index 13
0x3cb4-0x3cb7	NIC Diagnostic Return Ring Producer Index 14
0x3cb8-0x3cbb	NIC Diagnostic Return Ring Producer Index 15
0x3cbc-0x3cbf	NIC Diagnostic Return Ring Producer Index 16
0x3cc0-0x3cc3	NIC Diagnostic Send BD Consumer Index 1
0x3cc4-0x3cc7	NIC Diagnostic Send BD Consumer Index 2
0x3cc8-0x3ccb	NIC Diagnostic Send BD Consumer Index 3
0x3ccc-0x3ccf	NIC Diagnostic Send BD Consumer Index 4
0x3cd0-0x3cd3	NIC Diagnostic Send BD Consumer Index 5
0x3cd4-0x3cd7	NIC Diagnostic Send BD Consumer Index 6
0x3cd8-0x3cdb	NIC Diagnostic Send BD Consumer Index 7
0x3cdc-0x3cdf	NIC Diagnostic Send BD Consumer Index 8
0x3ce0-0x3ce3	NIC Diagnostic Send BD Consumer Index 9
0x3ce4-0x3ce7	NIC Diagnostic Send BD Consumer Index 10
0x3ce8-0x3ceb	NIC Diagnostic Send BD Consumer Index 11
0x3cec-0x3cef	NIC Diagnostic Send BD Consumer Index 12
0x3cf0-0x3cf3	NIC Diagnostic Send BD Consumer Index 13
0x3cf4-0x3cf7	NIC Diagnostic Send BD Consumer Index 14
0x3cf8-0x3cfb	NIC Diagnostic Send BD Consumer Index 15
0x3cfc-0x3cff	NIC Diagnostic Send BD Consumer Index 16
0x3d00-0x3fff	Reserved

Host Coalescing Mode Register (Offset 0x3C00)

Table 372: Host Coalescing Mode Register (Offset 0x3C00)

Bit	Field	Description	Init	Access
31-13	Reserved	-	0	R/W
12	No Interrupt on Force Update	When set, writing the Coalesce Now bit will cause a status without a corresponding interrupt event.		R/W
11	No Interrupt on DMAD Force	When set, the COAL_NOW bit of the buffer descriptor may be set to force a status block update without a corresponding interrupt (see "Send Buffer Descriptors" on page 94).		R/W
10	Clear Ticks Mode on TX	When set, the TX Host Coalescing Tick counter initializes to the idle state and begins counting only after a transmit BD event is detected.		R/W
9	Clear Ticks Mode on RX	When set, the RX Host Coalescing Tick counter initializes to the idle state and begins counting only after a receive BD event is detected.		R/W
8-7	Reserved (BCM5700, pre-C0 revision)	-		R/W
	Status Block Size (BCM5700 since C0 revision and rest of BCM57XX family)	Status Block Size for partial status block updates (BCM5700 MAC since C0 revision and the rest of the BCM57XX family, see "Status Block" on page 103): • 00: Full status block • 01: 64 byte • 10: 32 byte • 11: Undefined		R/W
6-4	MSI Bits	The least significant MSI 16-bit word is overwritten by these bits. Defaults to 0.		R/W
3	Coalesce Now	If set, Host Coalescing updates the Status Block immediately and sends an interrupt to host. This is a self-clearing bit. (For debug purpose only.)		R/W
2	Attn_Enable	When this bit is set to 1, an internal attention is generated when an error occurs.		R/W
1	Enable	This bit controls whether the Host Coalescing state machine is active or not. When set to 0, it completes the current operation and cleanly halts. Until it is completely halted, it remains one when read.		R/W
0	Reset	When this bit is set to 1, the Host Coalescing state machine is reset. This is a self-clearing bit.		R/W

Host Coalescing Status Register (Offset 0x3C04)

Table 373: Host Coalescing Status Register (Offset 0x3C04)

Bit	Field	Description	Init	Access
31-3	Reserved	-	0	R/O
2	Error	Host Coalescing error status.		R/O
1-0	Reserved	-	0	R/O

RECEIVE COALESCING TICKS REGISTERS (OFFSET 0X3C08)

The value in this register can be used to control how often the status block is updated (and how often interrupts are generated) due to receiving packets. The value in this register controls how many ticks, in units of 1 µs each, get loaded in an internal receive tick timer register. The timer will be reset to the value of this register and will start counting down after every status block update (regardless of the reason for the status block update). The timer is only reset after status block updates, and is not reset after any given packet is received. When the timer reaches 0, it will be considered to be in the expired state. Once the counter is in the expired state, a status block update will occur if a packet had been received and copied to host memory (via DMA) since the last status block update.

This register must be initialized by host software. A value of 0 in this register disables the receive tick coalescing logic. In this case, status block updates will occur for receive event only if the *Receive Max Coalesced BD* value is reached. Of course, status block updates for other reasons (e.g., transmit events) will also include any updates to the receive indices.

By setting the value in this register to a high number, a software device driver can reduce the number of status block updates and interrupts that occur due to receiving packets. This will generally increase performance in hosts that are under a high degree of stress and whose RISCs are saturated due to handling a large number of interrupts from the network controller. For host environments where receive interrupt latency needs to be very low, and the host is not close to be saturated, it is recommended that this register be set to 1.

SEND COALESCING TICKS REGISTER (OFFSET 0X3C0C)

The value in this register can be used to control how often the status block is updated (and how often interrupts are generated) according to the completion of transmit events. The value in this register controls how many ticks, in units of 1 µs each, get loaded in an internal transmit tick timer register. The timer will be reset to the value of this register and will start counting down, after every status block update (regardless of the reason for the status block update). The timer is only reset after status block updates, and is not reset after a transmit event completes. When the timer reaches 0, it will be considered to be in the expired state. Once the counter is in the expired state, a status block update will occur if a transmit event has occurred since the last status block update. In this case, a transmit event is defined by an update to one of the device's Send BD Consumer Indices. It should be noted that a Send Consumer Index increments whenever the data associated with a particular packet has been successfully moved (via DMA) across the bus, rather than when the packet is actually transmitted over the Ethernet wire.

This register must be initialized by host software. A value of 0 in this register disables the transmit tick coalescing logic. In this case, status block updates will occur for transmit events only if *the Send Max Coalesced BD* value is reached, or if the *BD_FLAG_COAL_NOW* bit is set in a send BD. Status block updates for other reasons (e.g., receive events) will also include any updates to the send indices.

By setting the value in this register to a high number, a software device driver can reduce the number of status block updates, and interrupts, that occur due to transmit completions. This will generally increase performance in hosts that do not require their send buffers to be freed quickly. For host environments that do require their send buffers to be recovered quickly, it is recommended that this register be set to 0.

Receive Max Coalesced BD Count (Offset 0x3C10)

This register contains the maximum number of receive return ring BDs that must filled in by the device before the device will update the status block due to a receive event.

Whenever the device completes the reception of a packet, it will fill in a receive return ring BD, and then increment an internal receive coalesce BD counter. When this internal counter reaches the value in this register, a status block update will occur. This counter will be reset (i.e., zeroed) whenever a status block update occurs regardless of the reason for the status block update.

This register must be initialized by host software. A value of 0 in this register disables the receive max BD coalescing logic. In this case, status block updates will occur for receive packets only via the *Receive Coalescing Ticks* mechanism. Status block updates for other reasons (e.g., transmit events) will also include any updates to the receive indices.

For simplicity, if a host wanted to get a status block update for every received packet, the host driver should just set this register to a value of 1. On the other hand, by setting the value in this register to a high number, a software device driver can reduce the number of status block updates and interrupts that occur due to receiving packets. This can increase performance in hosts that are under a high degree of stress and whose RISCs are saturated due to handling a large number of interrupts from the network controller. However, in lower traffic environments, there is no guarantee that consecutive packets will be received in a timely manner. Therefore, for those environments, it is recommended that *the Receive Coalescing Ticks* register are used to make sure that status block updates due to receiving packets are not delayed for an infinite amount of time.

Send Max Coalesced BD Count (Offset 0x3C14)

This register contains the maximum number of send BDs that must be processed by the device before the device will update the status block due to the transmission of packets.

Whenever the device completes the DMA of transmit packet buffer, it increments an internal send coalesce BD counter. When this internal counter reaches the value in this register, a status block update will occur. This counter will be reset (i.e. zeroed) whenever a status block update occurs regardless of the reason for the status block update.

This register must be initialized by host software. A value of 0 in this register disables the send max BD coalescing logic. In this case, status block updates will occur for receive packets only via the *Send Coalescing Ticks* mechanism. Of course, status block updates for other reasons (e.g., receive events) will also include any updates to the send indices.

For simplicity, if a host wanted to get a status block update for every transmitted packet, the host driver could just set this register to a value of 1. On the other hand, by setting the value in this register to a high number, a software device driver can reduce the number of status block updates and interrupts that occur due to transmitting packets. This can increase performance in hosts that are under a high degree of stress and whose RISCs are saturated due to handling a large number of interrupts from the network controller. However, in lower traffic environments, there is no guarantee that consecutive packets will be transmitted in a timely manner. Therefore, for those environments, it is recommended that the *Send Coalescing Ticks* register are used to make sure that status block updates due to transmitting packets are not delayed for an infinite amount of time.

RECEIVE COALESCING TICKS DURING INTERRUPT REGISTER (OFFSET 0X3C18)

This register is not applicable to BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 devices.

This register is very similar to the *Receive Coalescing Ticks* Register (see "Receive Coalescing Ticks Registers (Offset 0x3C08)" on page 453). However, this register is used instead of the *Receive Coalescing Ticks* register when the host is considered to be in its Interrupt Service Routine (ISR). In this case, the NIC considers the host to be in its ISR whenever either *Interrupt Mailbox 0* (see "Interrupt Mailbox 0 Register (Offset 0x200)" on page 372 for host standard and flat modes and "Interrupt Mailbox 0 Register (Offset 0x5800)" on page 492 for indirect mode) is set to a nonzero value, or when the *Mask Interrupt* bit is set.

When host software is in its ISR, this register is used to control the reset value of receive coalescing tick timer instead of the timer's reset value being determined by the non-during interrupt *Receive Coalescing Ticks* register. Furthermore, if the tick timer is set to the interrupt value in this register, and then later expires when the host is not in it ISR, then the timer is reset to the non-during interrupt value and re-evaluated.

If a host did not want to receive status block updates while its ISR, then the host driver should set this register to 0. If the host desired an immediate status block update after a packet was received and DMAed to the host, then the host driver could set this register to 1. If the host wanted delayed status block updates while in its ISR, then this register could be set to a value greater then 1.

SEND COALESCING TICKS DURING INTERRUPT REGISTER (OFFSET 0X3C1C)

This register is not applicable to BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 devices.

This register is very similar to the *Send Coalescing Ticks* Register (see "Send Coalescing Ticks Register (Offset 0x3C0C)" on page 453). However, this register is used instead of the *Send Coalescing Ticks* register when the host is considered to be in its interrupt service routine (ISR). In this case, the NIC considers the host to be in its ISR whenever either *Interrupt Mailbox 0* (see "Interrupt Mailbox 0 Register (Offset 0x200)" on page 372 for host standard and flat modes and "Interrupt Mailbox 0 Register (Offset 0x5800)" on page 492 indirect mode) is set to a nonzero value, or the *Mask Interrupt* bit is set.

When host software is in its ISR, this register is used to control the reset value of transmit coalescing tick timer instead of the timer's reset value being determined by the non-during interrupt *Send Coalescing Ticks* register. Furthermore, if the tick timer was set to the during interrupt value in this register, and then later expires when the host is not in it ISR, then the timer is reset to the non-during interrupt value and re-evaluated.

If a host did not want to receive status block updates while its ISR, then the host driver should set this register to 0. If the host desired an immediate status block update after a Send BD DMA completion, then the host driver could set this register to 1. If the host wanted delayed status block updates while in its ISR, then this register could be set to a value greater then 1.

Receive Max Coalesced BD Count During Interrupt (Offset 0x3C20)

This register is not applicable to BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 devices.

This register is very similar to the *Receive Max Coalesced BD Count* register. However, it is used instead of the *Receive Max Coalesced BD Count* register when the host is considered to be in its ISR. In this case, the NIC considers the host to be in its ISR whenever either *Interrupt Mailbox 0* (see "Interrupt Mailbox 0 Register (Offset 0x200)" on page 372 for host standard and flat modes and "Interrupt Mailbox 0 Register (Offset 0x5800)" on page 492 for indirect mode) is set to a nonzero value, or the *Mask Interrupt* bit is set.

When host software is in its ISR, this register is used instead of the *Receive Max Coalesced BD Count* register to determine how many receive return ring BDs must be completed before a status block is written back.

If a host did not want to get status block updates while its ISR, then the host driver should set this register to 0. If the host desired an immediate status block update after a packet was received and DMAed to the host, then the host driver could set this register to 1. If the host, while in it ISR, only wanted status block updates after handling multiple BDs or packets, then this register could be set to a value greater then 1.

Send Max Coalesced BD Count During Interrupt (Offset 0x3C24)

This register is not applicable to BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 devices.

This register is very similar to the *Send Max Coalesced BD Count* register. However, this register is used instead of the *Send Max Coalesced BD Count* register when the host is considered to be in its interrupt service routine (ISR). In this case, the NIC considers the host to be in its ISR whenever either *Interrupt Mailbox 0* (see "Interrupt Mailbox 0 Register (Offset 0x200)" on page 372 for host standard and flat modes and "Interrupt Mailbox 0 Register (Offset 0x5800)" on page 492 for indirect mode) is set to a nonzero value, or when the *Mask Interrupt* bit is set.

When host software is in its ISR, this register is used instead of the *Send Max Coalesced BD Count* register to determine how many send BDs must be processed by the hardware before a status block is written back.

If a host did not want to get status block updates while its ISR, then the host driver should set this register to 0. If the host desired an immediate status block update after a send BD was processed by the hardware, then the host driver could set this register to 1. If the host, while in it ISR, only wanted status block updates after handling multiple BDs or packets, then this register could be set to a value greater then 1.

Statistics Ticks Counter Register (Offset 0x3C28)

The Statistics Ticks register contains the number of clock ticks (of 1 µs each) that must elapse before the NIC DMAs the statistics block to the host. If set to zero then statistics are never DMAed to the host. This register must be initialized by the host. This register is not applicable to BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 devices.

Statistics Host Address Register (Offset 0x3C30)

This 64-bit register is in host address format and tells the NIC where to DMA the statistics block. This register is not applicable to BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 devices.

STATUS BLOCK HOST ADDRESS REGISTER (OFFSET 0X3C38)

This 64-bit register is in host address format and tells the NIC where to DMA the status block.

STATISTICS BASE ADDRESS REGISTER (OFFSET 0X3C40)

This 32-bit register is the location of the statistics structure in NIC memory. This register is not applicable to BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 devices.

STATUS BLOCK BASE ADDRESS REGISTER (OFFSET 0X3C44)

This 32-bit register is the location of the status block structure in NIC memory.

FLOW ATTENTION REGISTER (OFFSET 0X3C48)

The Flow attention register reports attentions from the various transmit and receive state machines, flow-through queues and the MBUF allocator. Whenever one of these blocks detects an attention situation, it sets the appropriate bit in the Flow attention register. Refer to the state machine causing the attention to determine the exact cause. The attention bits are cleared by writing a one to the bit (W2C). If a bit is marked as fatal, it means that the associated state machine is halted, and that corrective action must be taken by a CPU.

Table 374: Flow Attention Register (Offset 0x3C48)

Bit	Field	Description	Init	Fatality
31	Send BD Initiator	The Send BD Initiator state machine has caused an attention.		Fatal
30	Send BD Completion	The Send BD Completion state machine has caused an attention.		Fatal
29	Send BD Ring Selector	The Send BD Ring Selector state machine has caused an attention.		Fatal
28	Send Data Initiator	The Send Data Initiator state machine has caused an attention.		Fatal
27	Send Data Completion	The Send Data Completion state machine has caused an attention.		Fatal
26-24	Reserved	-	0	Fatal
23	Recv BD Initiator	The Recv BD Initiator state machine has caused an attention.		Fatal
22	Recv BD Completion	The Recv BD Completion state machine has caused an attention.		Fatal
21	Recv List Placement	The Recv List Placement state machine has caused an attention.		Fatal
20	Recv List Selector	The Recv List Selector state machine has caused an attention.		Fatal
19	Recv Data and Recv BD Initiator	The Recv Data and Recv BD Initiator state machine has caused an attention.		Fatal
18	Recv Data Completion	The Recv Data Completion state machine has caused an attention.		Fatal

Table 374: Flow Attention Register (Offset 0x3C48) (Cont.)

Bit	Field	Description	Init	Fatality
17	RCB Incorrectly Configured	Set if one of the RCBs is incorrectly configured based on the whole configuration.		Fatal
16	DMA Completion Discard	The DMA Completion Discard state machine has caused an attention.		Fatal
15	Host Coalescing	The Host Coalescing state machine has caused an attention.		Fatal
14-8	Reserved	-	0	
7	Memory Arbiter	The Memory Arbiter has caused an attention.		Fatal
6	MBUF Low Water	The MBUF allocation state machine has reached the mbuf low water threshold.		Non-fatal
5-0	Reserved	-		

NIC Receive BD Consumer Index Registers (Offset 0x3C50-0x3C58)

These three registers are shared by the Receive BD Completion and the Receive Data and Receive BD Initiator state machines. They are used to keep track of the receive BDs that have been DMAed to the NIC.

Note: The equivalent on the send side are in mailboxes. This is because there is no equivalent to NIC-based send rings on the receive side.

NIC Diagnostic Return Rings Producer Index Registers 1-16 (Offset 0x3C80-0x3CBC)

These 16 registers keep track of the NIC local copy of the return rings producer index (not the host copy which is DMAed by the Host Coalescing engine to the host). They are shared between the Send BD Initiator and the Host Coalescing state machines.

Note: The programmer should not write to these registers—they are for internal use only to aid with debugging and diagnostics.

Table 375: NIC Return Rings Producer Index (Offset 0x3C80)

Bit	Field	Description	Init	Access
31-11	Reserved	-	0	R/O
10-0	NIC Return Rings Producer Index	NIC Return Rings Producer Index		R/W

NIC Diagnostic Send BD Consumer Index Registers 1-16 (Offset 0x3CC0-0x3CFC)

These 16 registers keep track of the NIC local copy of the send BD ring consumers (not the host copy which is DMAed by the Host Coalescing engine to the host). They are shared between the Send BD Initiator and the Host Coalescing state machines.

Note: The programmer should not write to these registers—they are for internal use only to aid with debugging and diagnostics.

Table 376: NIC Send BD Consumer Index (Offset 0x3CC0)

Bit	Field	Description	Init	Access
31-16	Reserved	-	0	R/O
15-0	NIC Send BD Consumer Index	NIC Send BD Consumer Index		R/W

MEMORY ARBITER REGISTERS

Table 377: Memory Arbiter Registers

Offset	Registers
0x4000-0x4003	Memory Arbiter Mode
0x4004-0x4007	Memory Arbiter Status
0x4008-0x400b	Memory Arbiter Trap Address Low
0x400c-0x400f	Memory Arbiter Trap Address High
0x4010-0x43ff	Reserved

MEMORY ARBITER MODE REGISTER (OFFSET 0X4000)

Table 378: Memory Arbiter Mode Register (Offset 0x4000)

Bit	Field	Description	Init	Access
31-30	Tx MBUF Configuration (BCM5714 and BCM5715 only)	• 00 = 3 MBUFs for CPU • 01 = 13 MBUFs for CPU • 10 = 18 MBUFs for CPU • 11 = 35 MBUFs for CPU	01	R/W
29	CPU pipeline request disable (BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	CPU pipeline request disable. When set to 1, the write/read requests from the internal CPU will be processed sequentially (i.e., no back to back data valid).		R/W
	Reserved (other devices)	-	0	R/O
28	Low Latency Enable (BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	Low Latency Enable. • When set to 1, the read from the CPU to the RXMBUF will take the original MA protocol, where data_rd_valid always goes after cmd_ack. • If set to 0, the data_rd_valid overlaps at the same clock cycle as the cmd_ack.	0	R/W
	Reserved (other devices)	-	0	R/O
27	Fast Path Read Disable (BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	Fast Path Read Disable. When set to 1, the read from the CPU to the RXMBUF will take the slow path that goes through the original memory arbitration logic.		R/W
	Reserved (other devices)	-	0	R/O
26	Reserved	-	0	R/W

Table 378: Memory Arbiter Mode Register (Offset 0x4000) (Cont.)

Bit	Field	Description	Init	Access
25	Reserved (BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	-	0	R/W
	Ext SSRAM Data Read PAR ERR Enable (other devices)	External SSRAM data read parity error enable.		R/W
24	Reserved(BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	-	0	R/W
	Ext SSRAM WFIFO Underrun Enable (other devices)	External SSRAM Write FIFO Underrun enable.		R/W
23	Reserved (BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	-	0	R/W
	Ext SSRAM WFIFO Overrun Enable (other devices)	External SSRAM Write FIFO Overrun enable.		R/W
22	Reserved (BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	-	0	R/W
	Ext SSRAM RFIFO Underrun Enable (other devices)	External SSRAM Read FIFO Underrun enable R/W 21 Ext SSRAM RFIFO Overrun.		R/W
21	Reserved (BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	-	0	R/W
	Ext SSRAM RFIFO Overrun Enable (other devices)	External SSRAM Read FIFO Overrun enable.		R/W
20	DMAW2 Addr Trap Enable	DMA Write 2 Memory Arbiter request trap enable.		R/W
19	Reserved (BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	-	0	R/W
	BM Addr Trap Enable (other devices)	Buffer Manager Memory Arbiter request trap enable.		R/W

Table 378: Memory Arbiter Mode Register (Offset 0x4000) (Cont.)

Bit	Field	Description	Init	Access
18	Reserved (BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	-	0	R/W
	SBDS Addr Trap Enable (other devices)	Send BD Ring Selector Memory Arbiter request trap enable.		R/W
17	Reserved (BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	-	0	R/W
	SDC_DMAC Group Addr Trap Enable (other devices)	SDC_DMAC Group Memory Arbiter request trap enable.		R/W
16	SDI Addr Trap Enable	Send Data Initiator Memory Arbiter request trap enable.		R/W
15	Reserved (BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	-	0	R/W
	MCF Addr Trap Enable (other devices)	MBUF Cluster Free Memory Arbiter request trap enable.		R/W
14	Reserved (BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only))	-	0	R/W
	HC Addr Trap Enable (other devices)	Host Coalescing Memory Arbiter request trap enable.		R/W
13	Reserved (BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	-	0	R/W
	DC Group Addr Trap Enable (other devices)	DC Group Memory Arbiter request trap enable.		R/W
12	RDI2 Addr Trap Enable	Receive Data Initiator 2 Memory Arbiter request trap enable.		R/W
11	RDI1 Addr Trap Enable	Receive Data Initiator 1 Memory Arbiter request trap enable.		R/W
10	RQ Addr Trap Enable	Receive List Placement Memory Arbiter request trap enable.		R/W
9	Reserved (BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	-	0	R/W
	DMAR2 Addr Trap Enable (other devices)	DMA Read 2 Memory Arbiter request trap enable.		R/W
8	PCI Addr Trap Enable	PCI Memory Arbiter request trap enable.		R/W

Table 378: Memory Arbiter Mode Register (Offset 0x4000) (Cont.)

Bit	Field	Description	Init	Access
7	Reserved (BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	-	0	R/W
	TX RISC Addr Trap Enable (other devices)	TX RISC Memory Arbiter request trap enable.		R/W
6	RX RISC Addr Trap Enable	RX RISC Memory Arbiter request trap enable.		R/W
5	DMAR1 Addr Trap Enable	DMA Read 1 Memory Arbiter request trap enable.		R/W
4	DMAW1 Addr Trap Enable	DMA Write 1 Memory Arbiter request trap enable.		R/W
3	RX-MAC Addr Trap Enable	Receive MAC Memory Arbiter request trap enable.		R/W
2	TX-MAC Addr Trap Enable	Transmit MAC Memory Arbiter request trap enable.		R/W
1	Enable	This bit controls whether the Memory Arbiter is active or not. When set to 0, it completes the current operation and cleanly halts. Until it is completely halted, it remains 1 when read.		R/W
0	Reset	When this bit is set to 1, the Memory Arbiter state machine is reset. This is a self-clearing bit.		R/W

MEMORY ARBITER STATUS REGISTER (OFFSET 0X4004)

Table 379: Memory Arbiter Status Register (Offset 0x4004)

Bit	Field	Description	Init	Access
31-26	Reserved	-	0	R/O
25	Reserved(BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	-	0	R/W
	Ext SSRAM Data Read PAR ERR (other devices)	External SSRAM data read parity error.		W2C
24	Reserved (BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	-	0	R/W
	Ext SSRAM WFIFO Overrun (other devices)	External SSRAM Write FIFO Overrun.		W2C
23	Reserved (BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	-	0	R/W
	Ext SSRAM WFIFO Overrun (other devices)	External SSRAM Write FIFO Overrun.		W2C
22	Reserved (BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	-	0	R/W
	Ext SSRAM RFIFO Underrun (other devices)	External SSRAM Read FIFO Underrun.		W2C

Table 379: Memory Arbiter Status Register (Offset 0x4004) (Cont.)

Bit	Field	Description	Init	Access
21	Reserved (BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	-	0	R/W
	Ext SSRAM RFIFO Overrun	External SSRAM Read FIFO Overrun.		W2C
20	DMAW 2 Addr Trap (other devices)	DMA Write 2 Memory Arbiter request trap.		W2C
19	Reserved (BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	-	0	R/W
	BM Addr Trap (other devices)	Buffer Manager Memory Arbiter request trap.		W2C
18	Reserved (BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	-	0	R/W
	SBDS Addr Trap (other devices)	Send BD Ring Selector Memory Arbiter request trap.		W2C
17	Reserved (BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	-	0	R/W
	SDC_DMAC Group Addr Trap	SDC_DMAC Group Memory Arbiter request trap.		W2C
16	SDI Addr Trap (other devices)	Send Data Initiator Memory Arbiter request trap.		W2C
15	Reserved (BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	-	0	R/W
	MCF Addr Trap (other devices)	MBUF Cluster Free Memory Arbiter request trap.		W2C
14	Reserved (BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	-	0	R/W
	HC Addr Trap (other devices)	Host Coalescing Memory Arbiter request trap.		W2C
13	Reserved (BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	-	0	R/W
	DC Group Addr Trap (other devices)	DC Group Memory Arbiter request trap.		W2C
12	RDI2 Addr Trap	Receive Data Initiator 2 Memory Arbiter request trap.		W2C
11	RDI1 Addr Trap	Receive Data Initiator 1 Memory Arbiter request trap.		W2C
10	RQ Addr Trap	Receive List Placement Memory Arbiter request trap.		W2C
9	Reserved (BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	-	0	R/W
	DMAR2 Addr Trap (other devices)	DMA Read 2 Memory Arbiter request trap.		W2C
8	PCI Addr Trap	PCI Memory Arbiter request trap.		W2C
7	Reserved (BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	-	0	R/W
	TX RISC Addr Trap (other devices)	TX RISC Memory Arbiter request trap.		W2C

Table 379: Memory Arbiter Status Register (Offset 0x4004) (Cont.)

Bit	Field	Description	Init	Access
6	RX RISC Addr Trap	RX RISC Memory Arbiter request trap.		W2C
5	DMAR1 Addr Trap	DMA Read 1 Memory Arbiter request trap.		W2C
4	DMAW 1 Addr Trap	DMA Write 1 Memory Arbiter request trap.		W2C
3	RX-MAC Addr Trap	Receive MAC Memory Arbiter request trap.		W2C
2	TX-MAC Addr Trap	Transmit MAC Memory Arbiter request trap.		W2C
1-0	Reserved	-		R/O

MEMORY ARBITER TRAP ADDRESS LOW REGISTER (OFFSET 0X4008)

Table 380: Memory Arbiter Trap Address Low Register (Offset 0x4008)

Bit	Field	Description	Init	Access
31-21	Reserved	-	0	R/O
20-0	MA Trap Addr Low	Memory Arbiter Trap Address Low.		R/W

MEMORY ARBITER TRAP ADDRESS HIGH REGISTER (OFFSET 0X400C)

Table 381: Memory Arbiter Trap Address High Register (Offset 0x400C)

Bit	Field	Description	Init	Access
31-21	Reserved	-	0	R/O
20-0	MA Trap Addr High	Memory Arbiter Trap Address High.		R/W

BUFFER MANAGER CONTROL REGISTERS

Table 382: Buffer Manager Control Registers

Offset	*Registers*
0x4400-0x4403	Buffer Manager Mode register
0x4404-0x4407	Buffer Manager Status register
0x4408-0x440b	MBUF pool base address
0x440c-0x440f	MBUF pool length
0x4410-0x4413	MBUF pool Read DMA low watermark
0x4414-0x4417	MBUF pool MAC RX low watermark
0x4418-0x441b	MBUF pool high watermark
0x441c-0x441f	RX RISC MBUF Allocation Request register
0x4420-0x4423	RX RISC MBUF Allocation Response register
0x4424-0x4427	Reserved
0x4428-0x442b	Reserved
0x442c-0x442f	DMA Descriptor pool base address
0x4430-0x4433	DMA Descriptor pool length
0x4434-0x4437	DMA Descriptor pool low watermark
0x4438-0x443b	DMA Descriptor pool high watermark
0x443c-0x443f	Reserved
0x4440-0x4443	Reserved
0x4444-0x4447	Reserved
0x4448-0x444b	Reserved
0x444c-0x444f	Hardware Diagnostic 1 register
0x4450-0x4453	Hardware Diagnostic 2 register
0x4454-0x4457	Hardware Diagnostic 3 register
0x4458-0x445b	Receive Flow Threshold register
0x445c-0x47ff	Reserved

Buffer Manager Mode Register (Offset 0x4400)

Table 383: Buffer Manager Mode Register (Offset 0x4400)

Bit	Field	Description	Init	Access
31-6	Reserved	-	0	R/O
5	Reset RXMBUF Ptr (BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	When this bit is set, it will cause the RXMBUF allocation and deallocation pointer to reset back to the RXMBUF base. It will also cause the RXMAC to drop the preallocated MBUF and request a new one.	0	R/WC
	Reserved (other devices)	-	0	R/O
4	MBUF Low Attn Enable	MBUF Low Attn Enable MBUF low attention enable.		R/W
3	BM Test Mode	Buffer Manager Test Mode. Must be set to 0 for normal operation.		R/W
2	Attn_Enable	When this bit is set to 1, an internal attention is generated when an error occurs.		R/W
1	Enable	This bit controls whether the Buffer Manager is active or not. When set to 0 it completes the current operation and cleanly halts. Until it is completely halted, it remains one when read.		R/W
0	Reset	When this bit is set to 1, the Buffer Manager state machine is reset. This is a self-clearing bit.		R/W

Buffer Manager Status Register (Offset 0x4404)

Table 384: Buffer Manager Status Register (Offset 0x4404)

Bit	Field	Description	Init	Access
31-5	BM Test Mode	-		R/O
4	MBUF Low Attn	MBUF low attention status		R/O
3	Reserved	-		R/O
2	Error	Buffer Manager error status		R/O
1-0	Reserved	-	0	R/O

MBUF Pool Base Address Register (Offset 0x4408)

The MBUF Pool Base Address specifies beginning of the MBUF. This register can point to either:

- Internal memory (BCM5705, BCM5714, BCM5721, and BCM5751 only; see Table 385).
- Internal memory (rest of BCM57XX family except BCM5700; see Table 386).

BCM5705, BCM5714, BCM5721, and BCM5751 MAC Transceivers Only

This version of the MBUF Pool Base Address register applies to the BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 devices only.

Table 385: MBUF Pool Base Address Register (Offset 0x4408)

Bit	Field	Description	Init	Access
31-24	Reserved	-	0	R/O
23-0	mbuf Base Address	Specifies beginning of the MBUF for receive packet. The base address will ignore the lower seven bits, thus aligning the beginning of the MBUF pool on a 128-byte boundary.	10000h	R/W

Note: After modifying the value in this register, software should clear the contents of the RXMBUF memory and set the Reset RXMBUF Pointer bit of the Buffer Manger Mode register (offset 0x4400).

Rest of BCM57XX Family

This version of the MBUF Pool Base Address register applies to the rest of the BCM57XX family (except for the BCM5700 MAC).

Table 386: MBUF Pool Base Address Register (Offset 0x4408, Rest of BCM57XX Family)

Bit	Field	Description	Init	Access
31-0	MBUF Base Address	Specifies beginning of the MBUF. The base address will ignore the lower seven bits, thus aligning the beginning of the MBUF pool on a 128-byte boundary.	0	R/W

MBUF Pool Length Register (Offset 0x440C)

This 32-bit register specifies the length of MBUF (see Table 387 for the BCM5705, BCM5714, BCM5721, and BCM5751 and Table 388 for the rest of the BCM57XX family).

BCM5705, BCM5714, BCM5721, and BCM5751 MAC Transceivers Only

This version of the MBUF Pool Length Address register applies to the BCM5705, BCM5714, BCM5721, and BCM5751 MAC Transceivers only.

Table 387: MBUF Pool Length Register (Offset 0x440C)

Bit	Field	Description	Init	Access
31-24	Reserved	-	0	R/O
23-0	MBUF Length	Specifies the length of MBUF assigned for receive packet. The default is 32 KB. The lower seven bits should be ignored to align the MBUF pool on a 128-byte boundary.	0x8000	R/W

Note: After modifying the value in this register, software should clear the contents of the RXMBUF memory and set the Reset RXMBUF Pointer bit of the Buffer Manger Mode register (offset 0x4400).

Rest of BCM57XX Family

This version of the MBUF Pool Length Address register applies to the rest of the BCM57XX family (except for the BCM5700 MAC).

Table 388: MBUF Pool Length Register (Offset 0x440C, Rest of BCM57XX Family)

Bit	Field	Description	Init	Access
31-0	MBUF Length	Specifies length of MBUF. The length register can be up to 8 MB if there is external memory (only for the BCM5700) available to support this size MBUF pool. It is invalid to point to areas outside of memory.	0	R/W

Read DMA MBUF Low Watermark Register (Offset 0x4410)

This 32-bit register indicates the number of free MBUFs that must be available for the Read DMA Engine to dequeue a descriptor from the normal priority FTQ. If the free MBUF count drops below this mark, it must go above the high watermark to resume normal operation.

MAC RX MBUF Low Watermark Register (Offset 0x4414)

This 32-bit register indicates the number of free MBUFs that must be available for the RX MAC to accept a frame. If the free MBUF count drops below this mark, it must go above the high watermark to resume normal operation.

Note: When the MAC RX MBUF Low watermark has been reached, the RX MAC continues to accept incoming frames as configured by the "Low Watermark Maximum Receive Frames Register (Offset 0x504)" on page 394. If these additional incoming frames cause the MBUF free count to drop to 0, the Buffer Manager may stall and require a controller reset to recover.

MBUF High Watermark Register (Offset 0x4418)

This 32-bit register indicates the number of free MBUFs that must be available before normal operation is restored to the Read DMA Engine and/or the RX MAC.

RX RISC MBUF Cluster Allocation Request Register (Offset 0x441C)

The RX RISC MBUF Cluster Allocation Request register contains two fields:

- A requested size field which can be up to 64 KB long
- An allocation bit

The allocation bit is used to control the access to the response register. Use this register to set the size and allocation bit and then poll the register until the allocation bit is cleared. When the allocation bit is cleared, it is safe to read from the RX RISC MBUF Cluster Allocation Response register.

Table 389: RX RISC MBUF Allocation Request Register (Offset 0x441C)

Bit	Field	Description	Init	Access
31	Allocation Bit	Set this bit to 1 to request for the MBUF. When this bit is read as 0, then read the MBUF Allocation Response register (see "RX RISC MBUF Allocation Response Register (Offset 0x4420)" on page 471) for the TXMBUF pointer.	0	R/W
30-16	Reserved	-	0x0	R/O

15-0	Requested MBUF Cluster Size (BCM5700, BCM5701, BCM5702, BCM5703, and BCM5704 only)	This field controls the size of the MBUF cluster allocation and must include sufficient space for MBUF headers (8 bytes per MBUF) and frame descriptors (40 bytes per MBUF cluster).
		Broadcom suggests the following formula:
		`IF (packet_size <= 80)`
		`  request_size = packet_size + 48`
		`ELSE`
		`  request_size = packet_size + 48 + (8`
		`*(packet_size -48) / 120)`
			0	R/W

	Reserved	-	0	R/O

RX RISC MBUF Allocation Response Register (Offset 0x4420)

This register returns the MBUF cluster pointer of the specified size when the Allocation bit is cleared. If a second MBUF cluster allocation request is made before this register is read, an MBUF memory leak may occur. (For BCM5700, BCM5701, BCM5702, BCM5703, and BCM5704 only.)

This register is hardwired to 61, or 0x0000003D. The TXMBUF that is dedicated for ASF is the uppermost 384 bytes. The CPU should use 0x00009E80 as the starting address for ASF. (Applies to BCM5705, BCM5721, and BCM5751 only.)

RX CPU MBUF Allocation Response Register (0x4420h, BCM5714 and BCM5715 Only)

Table 390: RX CPU MBUF Allocation Response Register (0x4420h, BCM5714 and BCM5715 Only)

Bit	Field	Description	Init	Access
31:0	Tx MBUF Number	TX MBUF allocated to RX CPU. The response depends on the value bits 31:30 of Register 4000H (Memory arbiter mode register). • 0xAD if 4000[31:30] == 00 • 0xA3 if 4000[31:30] == 01 • 0x9E if 4000[31:30] == 10 • 0x8D if 4000[31:30] == 11	0xA3	R

TX RISC MBUF Allocation Response Register (Offset 0x4424)

This register is for BCM5700, BCM5701, BCM5702, BCM5703, and BCM5704 only. The TX RISC MBUF Cluster Allocation Request register contains two fields:

- A requested size field which can be up to 64 KB long
- An allocation bit

The allocation bit is used to control the access to the response register. Use this register to set the size and allocation bit and then poll the register until the allocation bit is cleared. When the allocation bit is cleared, it is safe to read from the TX RISC MBUF Cluster Allocation Response register.

Table 391: TX RISC MBUF Allocation Request Register (Offset 0x4424)

Bit	Field	Description	Init	Access
31	Allocation Bit	Set this bit to 1 to request for the MBUF. When this bit is read as 0, then read the MBUF Allocation Response register (see "RX RISC MBUF Allocation Response Register (Offset 0x4420)" on page 471) for the TXMBUF pointer.	0	R/W
30-16	Reserved	-	0x00000000	R/W

15-0	Requested MBUF Cluster Size	This field controls the size of the MBUF cluster allocation, and must include sufficient space for MBUF headers (8 bytes per MBUF) and frame descriptors (40 bytes per MBUF cluster).
		Broadcom suggests the following formula:
		`IF (packet_size <= 80)`
		`  request_size = packet_size + 48`
		`ELSE`
		`  request_size = packet_size + 48 + (8`
		`*(packet_size -48) / 120)`
			0	R/W

TX RISC MBUF Allocation Response Register (Offset 0x4428)

This register is for BCM5700, BCM5701, BCM5702, BCM5703, and BCM5704 only. This register returns the MBUF cluster pointer of the specified size when the Allocation bit is cleared. If a second MBUF cluster allocation request is made before this register is read, an MBUF memory leak may occur.

This register is hardwired to 61 or 0x0000003D. The TXMBUF that is dedicated for ASF is the uppermost 384 bytes. The CPU should use 0x00009E80 as the starting address for ASF.

DMA Descriptor Pool Initialization Register (Offset 0x442C-0x4433)

This register is not applicable to BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 devices.

There are two registers used to configure the DMA Descriptor pool. A base address and a length register. The base address register can only point to internal memory. The base address will ignore the lower five bits thus aligning the beginning of the DMA Descriptor pool on a 32-byte boundary. The length register can be up to 8 KB if there is memory available to support this size DMA Descriptor pool. It is invalid to point to areas outside of memory.

DMA Descriptor Pool Low Watermark Register (Offset 0x4434)

This register is not applicable to BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 devices.

This 32-bit register indicates the number of free DMA descriptors that must be available for the Send Data Initiator and Send BD Initiator state machines to generate DMAs.

DMA Descriptor Pool High Watermark Register (Offset 0x4438)

This register is not applicable to BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 devices.

This 32-bit register indicates the number of free DMA descriptors that must be available before normal operation is restored to the Send Data Initiator and Send BD Initiator state machines.

BM Hardware Diagnostic 1 Register (Offset 0x444C)

This 32-bit register provides debugging information on the TXMBUF pointer.

BCM5705, BCM5788, BCM5721, BCM5751, and BCM5752 Devices Only

This version of the BM Hardware Diagnostic 1 register applies to the BCM5705, BCM5788, BCM5721, BCM5751, and BCM5752 devices only.

Table 392: BM Hardware Diagnostic 1 Register (Offset 0x444C)

Bit	Field	Description	Init	Access
31-26	Reserved	-	0	R/W
25-20	Last TXMBUF Deallocation Head Pointer	Captures the last deallocation head pointer of the TXMBUF.	000000	R/O
19-16	Reserved	-	0000	R/W
15-10	Last TXMBUF Deallocation Tail Pointer	Captures the last deallocation head pointer of the TXMBUF.	000000	R/O
9-6	Reserved	-	0000	R/W
5-0	Next TXMBUF Allocation Pointer	The value of the next TXMBUF allocation pointer (should be between 0 and 60).	000000	R/O

BCM5714 and BCM5715 Devices Only

Table 393: Hardware Diagnostic 1 Register (0x444Ch, 5714 only)

Bits	Field	Description	Init	Access
31:28	Reserved	-	0	R
27:20	TX MBUF de-allocation pointer	TX de-allocation head pointer	-	R
19:18	Reserved	-	0	R
17:10	TX MBUF de-allocation pointer	TX de-allocation tail pointer	-	R
9:8	Reserved	-	0	R
7:0	TX MBUF Allocation pointer	TX allocation pointer	-	R

Rest of BCM57XX Family

This version of the BM Hardware Diagnostic 1 register applies to the rest of the BCM57XX family.

Table 394: BM Hardware Diagnostic 1 Register (Offset 0x444C)

Bit	Field	Description	Init	Access
31-0	Reserved	-	0	R/O

BM Hardware Diagnostic 2 Register (Offset 0x4450)

This 32-bit register provides debug information on the TXMBUF and RXMBUF counts.

BCM5705, BCM5788, BCM5721, BCM5751, and BCM5752 Devices Only

This version of the BM Hardware Diagnostic 2 register applies to the BCM5705, BCM5788, BCM5721, BCM5751, and BCM5752 devices only.

Table 395: BM Hardware Diagnostic 2 Register (Offset 0x4450)

Bit	Field	Description	Init	Access
31-25	Reserved	-	0000000	R/O
24-16	RXMBUF Count	The number of RXMBUFs that were allocated.	000000000	R/O
15	Reserved	-	0	R/O
14-9	TXMBUF Count	The number of TXMBUFs that were allocated.	000000	R/O
8-0	RXMBUF Left	The number of free RXMBUFs.	000000000	R/O

BCM5714 and BCM5715 Devices Only

Table 396: Hardware Diagnostic 2 Register (0x4450h, 5714 only)

Bits	Field	Description	Init	Access
31:26	Reserved	-	0	R
25:17	RX MBUF count	The number of Rx Mbufs that were allocated.	-	R
16:9	TX MBUF count	The number of Tx Mbufs that were allocated.	-	R
8:0	RX MBUF count	The number of free Rx Mbufs.	-	R

Rest of BCM57XX Family

This version of the BM Hardware Diagnostic 2 register applies to the rest of the BCM57XX family.

Table 397: BM Hardware Diagnostic 2 Register (Offset 0x4450)

Bit	Field	Description	Init	Access
31-24	Reserved	-	0	R/O
23-16	DMA Count	Free DMA Descriptor Count	0	R/O
15-0	Free MBUF Count	The number of free MBUFs	0	R/O

BM Hardware Diagnostic 3 Register (Offset 0x4454)

This 32-bit register provides debug information on the RXMBUF pointer.

BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 Devices Only

This version of the BM Hardware Diagnostic 3 register applies to the BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 devices only.

Table 398: BM Hardware Diagnostic 3 Register (Offset 0x4454)

Bit	Field	Description	Init	Access
31-25	Reserved	-	0000000	R/O
24-16	Next RXMBUF Deallocation Pointer	The next RXMBUF that is to be deallocated.	000000000	R/O
15-9	Reserved	-	0000000	R/O
8-0	Next RXMBUF Allocation Pointer	The next RXMBUF that is to be allocated.	000000000	R/O

Rest of BCM57XX Family

This version of the BM Hardware Diagnostic 3 register applies to the rest of the BCM57XX family.

Table 399: BM Hardware Diagnostic 3 Register (Offset 0x4454)

Bit	Field	Description	Init	Access
31-16	MBUF Free List Head	-	0	R/O
15-0	MBUF Free List Tail	-	0	R/O

RECEIVE FLOW THRESHOLD REGISTER (OFFSET 0X4458)

This register is not applicable to the BCM5700 MAC or BCM5701 MAC Devices.

Table 400: Receive Flow Threshold Register (Offset 0x4458)

Bit	Field	Description	Init	Access
31-16	Reserved	-	0	R/O
15-0	MBUF threshold	Defines the integer number of MBUFs remaining before the receive MAC will drop received frames.	0	R/W

READ DMA CONTROL REGISTERS

Table 401: Read DMA Control Registers

Offset	Registers
0x4800-0x4803	Read DMA Mode
0x4804-0x4807	Read DMA Status
0x4808-0x4bff	Reserved

READ DMA MODE REGISTER (OFFSET 0X4800)

Table 402: Read DMA Mode Register (Offset 0x4800)

Bit	Field	Description	Init	Access
31-30	Reserved (BCM5700, BCM5701, BCM5721, BCM5751, and BCM5752 only)	–	0	R/O
	Priority (Other devices)	Sets the priority of the DMA read engine relative to the DMA write engine and MSI engine. Equal settings result in fair round-robin arbitration. • 00: Lowest • 01: Low • 10: High • 11: Highest	00	R/W
29-28	Reserved	–	0	R/O
27	Hardware Post-DMA Enable (BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	Enable hardware LSO post-DMA processing.	0	R/W
	Reserved (other devices)	–	0	R/O
26	Post-DMA Debug Enable	When this bit is set, the Send Data Completion state machine will be halted if the Post-DMA bit of the Send BD is set.	0	R/W
	Reserved	–	0	R/O
25-18	Reserved	–	0	R/O

Table 402: Read DMA Mode Register (Offset 0x4800) (Cont.)

Bit	*Field*	*Description*	*Init*	*Access*
17-16	PCI Request Burst Length (BCM5705 and BCM5788 only)	The two bits define the burst length that the RDMA read engine would request to the PCI block. • 00 = FIFO available • 01 = 64 • 10 = 128 • 11 = BCM5705 only: Long Burst of up to 4K bytes. This setting can only be used for 33Mhz PCI and Core Clock speed of 62.5Mhz. • 11 = BCM5788 only: Reserved	0	R/W
	PCI Request Burst Length (BCM5721, BCM5751, and BCM5752 only)	The two bits define the burst length that the RDMA read engine would request to the PCI block. • 00 = 128 • 01 = 256 • 10 = 512 • 11 = 4K Bytes when Slow Core Clock Mode bit (bit-19 of 0x74 register) is 0; otherwise 512 Bytes.	0	R/W
	Reserved (other devices)	–	0	R/O
15-13	Reserved	–	0	R/O
12	Multi-Split Reset (BCM5704C and BCM5704S only)	When set, the multiple-split state machines are reset. This bit is self-clearing.	0	R/W
	Reserved (other devices)	–	0	R/O
11	Multi-Split Enable (BCM5704C and BCM5704S only)	Multiple-split mode is enabled by writing a 1 to bit 10. ***Note:*** For multiple split mode to work, the PCIX target (IO bridge) will always have to respond with either a split response or a retry. The target cannot give data directly in response to a read request for multiple split mode to work. Also, the split completion data has to come back in order. This has to be documented in the register spec. Note this requirement is supported by CIOB.	0	R/W
	Reserved (other devices)	–	0	R/O
10	Read DMA PCI-X Split Transaction Timeout Expired Attention Enable	Enable read DMA PCI-X split transaction timeout expired attention.	0	R/W
9	Read DMA Local Memory Write Longer Than DMA Length Attention Enable	Enable Read DMA Local Memory Write Longer Than DMA Length Attention.	0	R/W
8	Read DMA PCI FIFO Overread Attention Enable	Enable Read DMA PCI FIFO Overread Attention (PCI read longer than DMA length.)	0	R/W
7	Read DMA PCI FIFO Underrun Attention Enable	Enable Read DMA PCI FIFO Underrun Attention.	0	R/W
6	Read DMA PCI FIFO Overrun Attention Enable	Enable Read DMA PCI FIFO Overrun Attention.	0	R/W
5	Read DMA PCI Host Address Overflow Error Attention Enable	Enable Read DMA PCI Host Address Overflow Error Attention. A host address overflow occurs when a single DMA read begins at an address below 4 GB and ends on an address above 4 GB. This is a fatal error.	0	R/W
4	Read DMA PCI Parity Error Attention Enable	Enable Read DMA PCI Parity Error Attention.	0	R/W

Table 402: Read DMA Mode Register (Offset 0x4800) (Cont.)

Bit	Field	Description	Init	Access
3	Read DMA PCI Master Abort Attention Enable	Enable Read DMA PCI Master Abort Attention.	0	R/W
2	Read DMA PCI Target Abort Attention Enable	Enable Read DMA PCI Target Abort Attention.	0	R/W
1	Enable	This bit controls whether the Read DMA state machine is active or not. When set to 0, it completes the current operation and cleanly halts. Until it is completely halted, it remains one when read.	1	R/W
0	Reset	When this bit is set to 1, the Read DMA state machine is reset. This is a self-clearing bit.	0	R/W

READ DMA STATUS REGISTER (OFFSET 0X4804)

Table 403: Read DMA Status Register (Offset 0x4804)

Bit	Field	Description	Init	Access
31-11	Reserved	-	0	R/O
10	Read DMA PCI-X Split Transaction Timeout Expired	Read DMA PCI-X split transaction timeout expired.	0	W2C
9	Read DMA Local Memory Write Longer Than DMA Length Error	Read DMA Local Memory Write Longer Than DMA Length Error.	0	W2C
8	Read DMA PCI FIFO Overread Error	Read DMA PCI FIFO Overread Error (PCI read longer than DMA length).	0	W2C
7	Read DMA PCI FIFO Underrun Error	Read DMA PCI FIFO Underrun Error.	0	W2C
6	Read DMA PCI FIFO Overrun Error	Read DMA PCI FIFO Overrun Error.	0	W2C
5	Read DMA PCI Host Address Overflow Error	Read DMA PCI Host Address Overflow Error. A host address overflow occurs when a single DMA read begins at an address below a multiple of 4 GB and ends at an address above the same multiple of 4 GB (i.e., the host memory address transitions from 0xXXXXXXXX_FFFFFFFF to 0xYYYYYYYY_00000000 in a single read). This is a fatal error.	0	W2C
4	Read DMA PCI Parity Error	Read DMA PCI Parity Error.	0	W2C
3	Read DMA PCI Master Abort Error	Read DMA PCI Master Abort Error.	0	W2C
2	Read DMA PCI Target Abort Error	Read DMA PCI Target Abort Error.	0	W2C
1-0	Reserved	-	0	W2C

WRITE DMA CONTROL REGISTERS

Table 404: Write DMA Control Registers

Offset	Registers
0x4c00-0x4c03	Write DMA Mode
0x4c04-0x4c07	Write DMA Status
0x4c08-0x4fff	Reserved

WRITE DMA MODE REGISTER (OFFSET 0X4C00)

Table 405: Write DMA Mode Register (Offset 0x4C00)

Bit	Field	Description	Init	Access
31-30	Reserved (BCM5700, BCM5701, BCM5721, BCM5751, and BCM5752 only)	-	0	R/O
	Priority (other devices)	Sets the priority of the DMA read engine relative to the DMA write engine and MSI engine. Equal settings result in fair round-robin arbitration. • 00 = Lowest • 01 = Low • 10 = High • 11 = Highest	00	R/W
29	Extended BD Enable (BCM5714 and BCM5715 only)	Set to 1 to enable the use of Extended BDs in Standard Receive Ring.	0	R/W
	Reserved (Rest of BCM57XX family)	-		
28-19	Reserved	-		R/O
18	Swap Test Enable (BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	When this bit is set, swap test mode will be enabled and bits 17 to 12 can be used to test different byte/word swap settings.	0	R/W
	Reserved (other devices)	-		R/O
17	HC Byte Swap (BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	Byte swap control for status words.	0	R/W
	Reserved (other devices)	-		R/O
16	HC Word Swap (BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	Word swap control for status words.	0	R/W
	Reserved (other devices)	-		R/O

Table 405: Write DMA Mode Register (Offset 0x4C00) (Cont.)

Bit	*Field*	*Description*	*Init*	*Access*
15	BD Byte Swap (BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	Byte swap control for return BDs.	0	R/W
	Reserved (other devices)	-		R/O
14	BD Word Swap (BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	Word swap control for return BDs.	0	R/W
	Reserved (other devices)	-		R/O
13	Data Byte Swap (BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	Byte swap control for data.	0	R/W
	Reserved (other devices)	-		R/O
12	Data Word Swap (BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	Word swap control for data.	0	R/W
	Reserved (other devices)	-		R/O
11	Software Byte Swap Control (BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	To override byte enables with all 1s.	0	R/W
	Reserved (other devices)	-		R/O
10	Reserved (other devices)	-		R/O
	Receive Accelerate Mode (BCM5705, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	The write DMA-to-PCI request length is the available data size in the PCI RX FIFO. Set to 1: The write DMA-to-PCI request length is the maximum length of the current transaction, regardless of the available data size in PCI RX FIFO. This mode cannot be used in slow core clock environment. Disable this mode before switching to slow core clock mode.	0	R/W
9	Write DMA Local Memory Read Longer Than DMA Length	Attention Enable. Enable Write DMA Local Memory Read Longer Than DMA Length Attention.	0	R/W
8	Write DMA PCI FIFO Overwrite Attention Enable	Enable Write DMA PCI FIFO Overwrite Attention (PCI write longer than DMA length).	0	R/W
7	Write DMA PCI FIFO Underrun Attention Enable	Enable Write DMA PCI FIFO Underrun Attention.	0	R/W
6	Write DMA PCI FIFO Overrun Attention Enable	Enable Write DMA PCI FIFO Overrun Attention.	0	R/W
5	Write DMA PCI Host Address Overflow Error Attention Enable	Enable Write DMA PCI Host Address Overflow Error Attention.	0	R/W
4	Write DMA PCI Parity Error Attention Enable	Enable Write DMA PCI Parity Error Attention.	0	R/W

Table 405: Write DMA Mode Register (Offset 0x4C00) (Cont.)

Bit	Field	Description	Init	Access
3	Write DMA PCI Master Abort Attention Enable	Enable Write DMA PCI Master Abort Attention.	0	R/W
2	Write DMA PCI Target Abort Attention Enable	Enable Write DMA PCI Target Abort Attention.	0	R/W
1	Enable	This bit controls whether the Write DMA state machine is active or not. When set to 0, it completes the current operation and cleanly halts. Until it is completely halted, it remains 1 when read.	1	R/W
0	Reset	When this bit is set to 1, the Write DMA state machine is reset. This is a self-clearing bit.	0	R/W

Write DMA Status Register (Offset 0x4C04)

Table 406: Write DMA Status Register (Offset 0x4C04)

Bit	Field	Description	Init	Access
31-10	Reserved	-	0	R/O
9	Write DMA Local Memory Read Longer Than DMA Length Error	Write DMA Local Memory Read Longer Than DMA Length Error.	0	W2C
8	Write DMA PCI FIFO Overread Error	Write DMA PCI FIFO Overread Error. (PCI read longer than DMA length)	0	W2C
7	Write DMA PCI FIFO Underrun Error	Write DMA PCI FIFO Underrun Error.	0	W2C
6	Write DMA PCI FIFO Overrun Error	Write DMA PCI FIFO Overrun Error.	0	W2C
5	Write DMA PCI Host Address Overflow Error	Write DMA PCI Host Address Overflow Error. A host address overflow occurs when a single DMA write begins at an address below a multiple of 4 GB and ends at an address above the same multiple of 4 GB (i.e., the host memory address transitions from 0xXXXXXXXX_FFFFFFFF to 0xYYYYYYYY_00000000 in a single write). This is a fatal error.	0	W2C
4	Write DMA PCI Parity Error	Write DMA PCI Parity Error.	0	W2C
3	Write DMA PCI Master Abort Error	Write DMA PCI Master Abort Error.	0	W2C
2	Write DMA PCI Target Abort Error	Write DMA PCI Target Abort Error.	0	W2C
1-0	Reserved	-	0	R/O

RX RISC REGISTERS

The following RX RISC registers are exposed to host software to provide a mechanism to download firmware binary. The information in this section is not intended to provide a comprehensive understanding of the RISC architecture.

Table 407: RX RISC Registers

Offset	*Register*
0x5000-0x5003	RX RISC Mode Register
0x5004-0x5007	RX RISC State Register
0x5008-0x501b	Reserved
0x501c-0x501f	RX RISC Program Counter
0x5020-0x5033	Reserved
0x5034-0x5037	RX RISC Hardware Breakpoint Register
0x5038-0x53ff	Reserved

RX RISC MODE REGISTER (OFFSET 0X5000)

This register controls the operation of the RX RISC and its miscellaneous functions.

Table 408: RX RISC Mode Register Fields (Offset 0x5000)

Bit	*Field*	*Description*	*Init*	*Access*
31-15	Reserved	Always 0.	0	R/O
14	Enable register address trap halt	When set, if the GRC raises the trap signal to this processor, it will halt. Cleared on reset and Watchdog interrupt.	0	RW
13	Enable memory address trap halt	When set, if the MA raises the trap signal to this processor, it will halt. Cleared on reset and Watchdog interrupt.	0	RW
12	Enable Invalid Instruction Fetch halt	When set, the condition that causes RX RISC state bit 6 to be set, also halts the RX RISC. Set by reset. Cleared by Watchdog interrupt.	0	RW
11	Enable Invalid Data access halt	When set, the condition that causes RX RISC state bit 5 to be set, also halts the RX RISC. Set by reset. Cleared by Watchdog interrupt.	0	RW
10	Halt RX RISC	Set by TX RISC or the host to halt the RX RISC. Cleared on reset and Watchdog interrupt.	0	RW
9	Flush Instruction Cache	Self-clearing bit which forces the instruction cache to flush.	0	WO
8	Enable Instruction Cache prefetch	Enables prefetch logic within the instruction cache. When disabled only a single cache line is read on a cache miss. Cleared on reset.	0	RW
7	Enable Watchdog	Enables watchdog interrupt state machine. Used in conjunction with Watchdog Clear register, Watchdog Saved PC register and Watchdog Vector register. Cleared on reset and Watchdog interrupt.	0	RW

Table 408: RX RISC Mode Register Fields (Offset 0x5000) (Cont.)

Bit	Field	Description	Init	Access
6	ROM Fail	Asserted on reset. Cleared by ROM code after it successfully loads code from NVRAM. Afterwards, this bit can be used by software for any purpose.	1	RW
5	Enable Data Cache	Enables the data cache. Cleared on reset. ***Note:*** Firmware developers should take care to clear this bit before polling internal SRAM memory locations, because the RX RISC processor uses a two-element LRU caching algorithm, which is not affected by writes from the PCI interface.	0	RW
4	Enable Write Post Buffers	Enables absorption of multiple SW operations for SRAM and register writes. When this bit is disabled, only one write at a time will be absorbed by the write post buffers. Cleared on reset. ***Note:*** Setting this bit on the BCM5705, BCM5721, and BCM5751 may cause unpredictable behavior.	0	RW
3	Enable Page 0 Instr Halt	When set, instruction references to the first 256 bytes of SRAM force the RX RISC to halt and cause bit 4 in the RX RISC state register to be latched. Cleared on reset and Watchdog interrupt.	0	RW
2	Enable Page 0 Data Halt	When set, data references to the first 256 bytes of SRAM force the RX RISC to halt and cause bit 3 in the RX RISC state register to be latched. Cleared on reset and Watchdog interrupt.	0	RW
1	Single-Step RX RISC	Advances the RX RISC's PC for one cycle. If halting condition still exists, the RX RISC will again halt; otherwise, it will resume normal operation.	0	RW
0	Reset RX RISC	Self-clearing bit which resets only the RX RISC.	0	WO

RX RISC STATE REGISTER (OFFSET 0X5004)

The RX RISC State register reports the current state of the RX RISC and, if halted, gives reasons for the halt. There are four categories of information; informational (read-only), informational (write-to-clear), disable-able halt conditions (write-to-clear), and non-disable-able halt conditions (write-to-clear).

Table 409: RX RISC State Fields (Offset 0x5004)

Bit	Field	Description	Init	Access
31	Blocking read	A blocking data cache miss occurred, causing the RX RISC to stall while data is fetched from external (to the RX RISC) memory. This is intended as a debugging tool. No state is saved other than the fact that the miss occurred.	0	W2C
30	MA request FIFO overflow	MA_req_FIFO overflowed. The RX RISC is halted on this condition.	0	W2C
29	MA data/bytemask FIFO overflow	MA_datamask_FIFO overflowed. The RX RISC is halted on this condition.	0	W2C
28	MA outstanding read FIFO overflow	MA_rd_FIFO overflowed. The RX RISC is halted on this condition.	0	W2C
27	MA outstanding write FIFO overflow	MA_wr_FIFO overflowed. The RX RISC is halted on this condition.	0	W2C
26-16	Reserved	Always 0.	0	R/O
15	Instruction fetch stall	The processor is currently stalled due to an instruction fetch.	0	R/O
14	Data access stall	The processor is currently stalled due to a data access.	0	R/O
13-11	Reserved	Always 0.	0	R/O
10	RX RISC Halted	The RX RISC was explicitly halted via bit 10 in the RX RISC Mode register.	0	R/O
9	Register address trap	A signal was received from the Global Resources block indicating that this processor accessed a register location that triggered a software trap. The GRC registers are used to configure register address trapping.	0	W2C
8	Memory address trap	A signal was received from the Memory Arbiter indicating that some BCM5700 block, possibly this processor, accessed a memory location that triggered a software trap. The MA registers are used to configure memory address trapping.	0	W2C
7	Bad Memory Alignment	Load or Store instruction was executed with the least significant two address bits not valid for the width of the operation (e.g., Load word or Load Half-word from an odd byte address).	0	W2C
6	Invalid Instruction Fetch	Program Counter (PC) is set to invalid location in processor address space. See "Memory Maps and Pool Configuration" on page 171 for details about unmapped areas in the CPU address space.	0	W2C
5	Invalid Data Access	Data reference to illegal location. See "Memory Maps and Pool Configuration" on page 171 for details about unmapped areas in the CPU address space.	0	W2C
4	Page 0 Instruction Reference	When enabled in mode register, indicates the address in the PC is within the lower 256 bytes of SRAM.	0	W2C

Table 409: RX RISC State Fields (Offset 0x5004) (Cont.)

Bit	Field	Description	Init	Access
3	Page 0 Data Reference	When enabled in mode register, indicates data reference within lower 256 bytes of SRAM.	0	W2C
2	Invalid Instruction	Invalid instruction fetched.	0	W2C
1	Halt Instruction Executed	A halt-type instruction was executed by the RX RISC.	0	W2C
0	Hardware Breakpoint	When enabled in mode register, indicates hardware breakpoint has been reached.	0	W2C

RX RISC Program Counter (Offset 0x501C)

The program counter register can be used to read or write the current Program Counter of the each CPU. Reads can occur at any time, however writes can only be performed when the CPU 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.

RX RISC Hardware Breakpoint Register (Offset 0x5034)

This register is used to set a hardware breakpoint based on the RISC's program counter (PC). If the PC equals the value in this register, and the hardware breakpoint is enabled, the RISC is halted and the appropriate stopping condition is indicated in the RISC State Register. To enable the hardware breakpoint, simply write the byte address of the instruction to break on and clear the Disable Hardware Breakpoint bit.

This register is also used to indicate the Progress Code for the BCM57XX ROM Loader.

Table 410: RX RISC Hardware Breakpoint Register (offset 0x5034)

Bit	Field	Description	Init	Access
31-2	Hardware Breakpoint	Word address to break on.	0	R/W
1	Reserved	-	0	R/O
0	Disable Hardware Breakpoint	When this bit is set, the Hardware Breakpoint is disabled.	1	R/W

TX RISC Registers

These registers are applicable to BCM5700, BCM5701, BCM5702, BCM5703, and BCM5704 only.

The following TX RISC registers are exposed to host software to provide a mechanism to download firmware binary. The information in this section is not intended to provide a comprehensive understanding of the RISC architecture. The reader should refer to the BCM57XX *Family Firmware Reference Manual* for an understanding of the RISC architecture.

Table 411: TX RISC Registers

Offset	*Register*
0x5400-0x5403	TX RISC Mode Register
0x5404-0x5407	TX RISC State Register
0x5408-0x541b	Reserved
0x541c-0x541f	TX RISC Program Counter
0x5420-0x57ff	Reserved

TX RISC Mode Register (Offset 0x5400)

Table 412: TX RISC Mode Register Fields (Offset 0x5400)

Bit	*Field*	*Description*	*Init*	*Access*
31-15	Reserved	Always 0.	0	R/O
14	Enable register address trap halt	When set, if the GRC raises the trap signal to this processor, it will halt. Cleared on reset and Watchdog interrupt.	0	RW
13	Enable memory address trap halt	When set, if the MA raises the trap signal to this processor, it will halt. Cleared on reset and Watchdog interrupt.	0	RW
12	Enable Invalid Instruction Fetch halt	When set, the condition that causes TX RISC state bit 6 to be set, also halts the TX RISC. Set by reset. Cleared by Watchdog interrupt.	0	RW
11	Enable Invalid Data access halt	When set, the condition that causes TX RISC state bit 5 to be set, also halts the TX RISC. Set by reset. Cleared by Watchdog interrupt.	0	RW
10	Halt TX RISC	Set by TX RISC or the host to halt the TX RISC. Cleared on reset and Watchdog interrupt.	0	RW
9	Flush Instruction Cache	Self-clearing bit, which forces the instruction cache to flush.	0	WO
8	Enable Instruction Cache prefetch	Enables prefetch logic within the instruction cache. When disabled only a single cache line is read on a cache miss. Cleared on reset.	0	RW
7	Enable Watchdog	Enables watchdog interrupt state machine. Used in conjunction with Watchdog Clear register, Watchdog Saved PC register and Watchdog Vector register. Cleared on reset and Watchdog interrupt.	0	RW

Table 412: TX RISC Mode Register Fields (Offset 0x5400) (Cont.)

Bit	Field	Description	Init	Access
6	ROM Fail	Asserted on reset. Cleared by ROM code after it successfully loads code from NVRAM. Afterwards, this bit can be used by software for any purpose.	1	RW
5	Enable Data Cache	Enables the data cache. Cleared on reset. ***Note:*** Firmware developers should take care to clear this bit before polling internal SRAM memory locations, because the TX RISC processor uses a two-element LRU caching algorithm, which is not affected by writes from the PCI interface.	0	RW
4	Enable Write Post Buffers	Enables absorption of multiple SW operations for SRAM and register writes. When this bit is disabled, only one write at a time will be absorbed by the write post buffers. Cleared on reset.	0	RW
3	Enable Page 0 Instr Halt	When set, instruction references to the first 256 bytes of SRAM force the TX RISC to halt and cause bit 4 in the TX RISC state register to be latched. Cleared on reset and Watchdog interrupt.	0	RW
2	Enable Page 0 Data Halt	When set, data references to the first 256 bytes of SRAM force the TX RISC to halt and cause bit 3 in the TX RISC state register to be latched. Cleared on reset and Watchdog interrupt.	0	RW
1	Single-Step TX RISC	Advances the TX RISC's PC for one cycle. If halting condition still exists, the TX RISC will again halt, otherwise it will resume normal operation.	0	RW
0	Reset TX RISC	Self-clearing bit which resets only the TX RISC.	0	WO

TX RISC State Register (Offset 0x5404)

The TX RISC State register reports the current state of the TX RISC and, if halted, gives reasons for the halt. There are four categories of information; informational (read-only), informational (write-to-clear), disable-able halt conditions (write-to-clear), and non-disable-able halt conditions (write-to-clear).

Table 413: TX RISC State Fields (Offset 0x5404)

Bit	Field	Description	Init	Access
31	Blocking read	A blocking data cache miss occurred, causing the TX RISC to stall while data is fetched from external (to the TX RISC) memory. This is intended as a debugging tool. No state is saved other than the fact that the miss occurred.	0	W2C
30	MA request FIFO overflow	MA_req_FIFO overflowed. The TX RISC is halted on this condition.	0	W2C
29	MA data/bytemask FIFO overflow	MA_datamask_FIFO overflowed. The TX RISC is halted on this condition.	0	W2C
28	MA outstanding read FIFO overflow	MA_rd_FIFO overflowed. The TX RISC is halted on this condition.	0	W2C
27	MA outstanding write FIFO overflow	MA_wr_FIFO overflowed. The TX RISC is halted on this condition.	0	W2C
26-16	Reserved	Always 0.	0	R/O

Table 413: TX RISC State Fields (Offset 0x5404) (Cont.)

Bit	Field	Description	Init	Access
15	Instruction fetch stall	The processor is currently stalled due to an instruction fetch.	0	R/O
14	Data access stall	The processor is currently stalled due to a data access.	0	R/O
13-11	Reserved	Always 0.	0	R/O
10	TX RISC Halted	The TX RISC was explicitly halted via bit 10 in the TX RISC Mode register.	0	R/O
9	Register address trap	A signal was received from the Global Resources block indicating that this processor, accessed a register location that triggered a software trap. The GRC registers are used to configure register address trapping.	0	W2C
8	Memory address trap	A signal was received from the Memory Arbiter indicating that some BCM5700 block, possibly this processor, accessed a memory location that triggered a software trap. The MA registers are used to configure memory address trapping.	0	W2C
7	Bad Memory Alignment	Load or Store instruction was executed with the least significant two address bits not valid for the width of the operation (e.g., Load word or Load Half-word from an odd byte address).	0	W2C
6	Invalid Instruction Fetch	Program Counter (PC) is set to invalid location in processor address space. See"Memory Maps and Pool Configuration" on page 171 for details about unmapped areas in the CPU address space.	0	W2C
5	Invalid Data Access	Data reference to illegal location. See "Memory Maps and Pool Configuration" on page 171 for details about unmapped areas in the CPU address space.	0	W2C
4	Page 0 Instruction Reference	When enabled in mode register, indicates the address in the PC is within the lower 256 bytes of SRAM.	0	W2C
3	Page 0 Data Reference	When enabled in mode register, indicates data reference within lower 256 bytes of SRAM.	0	W2C
2	Invalid Instruction	Invalid instruction fetched.	0	W2C
1	Halt Instruction Executed	A halt-type instruction was executed by the TX RISC.	0	W2C
0	Hardware Breakpoint	When enabled in mode register, indicates hardware breakpoint has been reached.	0	W2C

TX RISC PROGRAM COUNTER (OFFSET 0X541C)

The program counter register can be used to read or write the current Program Counter of the each CPU. Reads can occur at any time, however, writes can only be performed when the CPU 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.

Low-Priority Mailboxes

This is a 512 byte region that contains 64 registers. These mailbox registers are:

- 64 bits for the BCM5700 MAC and BCM5701 MAC Transceivers.
- 32 bits for the rest of the BCM57XX family.

These registers are called low-priority mailbox registers (or low-priority mailboxes). When a value is stored in the least significant 32 bits of these registers, an event (known as a Mailbox Event) is generated to the one of the RX RISC or TX RISC. To write 64 bits of a mailbox location, the upper 32 bits should be written to before the lower 32 bits.

In the BCM5702 and later devices, the upper 32 bits are not used. For compatibility across the BCM57XX family, access only the lower 32 bits.

Note: The low-priority mailbox registers are for indirect register access mode only. For host standard and flat access modes, access the mailboxes via the high-priority mailboxes (see "High-Priority Mailboxes" on page 370).

Table 414: Low-Priority Mailbox Structure

31	*15*	*0*	*Init*
Mailbox	-	-	0x00
Not used in the BCM5702 MAC Transceiver and later.			0x04

Table 415: Low-Priority Mailbox Registers

Offset	*Registers*
0x5800-0x5807	Interrupt Mailbox 0
0x5808-0x580f	Interrupt Mailbox 1
0x5810-0x5817	Interrupt Mailbox 2
0x5818-0x581f	Interrupt Mailbox 3
0x5820-0x5827	General Mailbox 1
0x5828-0x582f	General Mailbox 2
0x5830-0x5837	General Mailbox 3
0x5838-0x583f	General Mailbox 4
0x5840-0x5847	General Mailbox 5
0x5848-0x584f	General Mailbox 6
0x5850-0x5857	General Mailbox 7
0x5858-0x585f	General Mailbox 8
0x5860-0x5867	Reserved
0x5868-0x586f	Receive BD Standard Producer Ring Producer Index
0x5870-0x5877	Receive BD Jumbo Producer Ring Producer Index
0x5878-0x587f	Receive BD Mini Producer Ring Producer Index

Table 415: Low-Priority Mailbox Registers (Cont.)

Offset	*Registers*
0x5880-0x5487	Receive BD Return Ring 1 Consumer Index
0x5888-0x588f	Receive BD Return Ring 2 Consumer Index
0x5890-0x5897	Receive BD Return Ring 3 Consumer Index
0x5898-0x589f	Receive BD Return Ring 4 Consumer Index
0x58a0-0x58a7	Receive BD Return Ring 5 Consumer Index
0x58a8-0x58af	Receive BD Return Ring 6 Consumer Index
0x58b0-0x58b7	Receive BD Return Ring 7 Consumer Index
0x58b8-0x58bf	Receive BD Return Ring 8 Consumer Index
0x58c0-0x58c7	Receive BD Return Ring 9 Consumer Index
0x58c8-0x58cf	Receive BD Return Ring 10 Consumer Index
0x58d0-0x58d7	Receive BD Return Ring 11 Consumer Index
0x58d8-0x58df	Receive BD Return Ring 12 Consumer Index
0x58e0-0x58e7	Receive BD Return Ring 13 Consumer Index
0x58e8-0x58ef	Receive BD Return Ring 14 Consumer Index
0x58f0-0x58f7	Receive BD Return Ring 15 Consumer Index
0x58f8-0x58ff	Receive BD Return Ring 16 Consumer Index
0x5900-0x5907	Send BD Ring 1 Host Producer Index
0x5908-0x590f	Send BD Ring 2 Host Producer Index
0x5910-0x5917	Send BD Ring 3 Host Producer Index
0x5918-0x591f	Send BD Ring 4 Host Producer Index
0x5920-0x5927	Send BD Ring 5 Host Producer Index
0x5928-0x592f	Send BD Ring 6 Host Producer Index
0x5930-0x5937	Send BD Ring 7 Host Producer Index
0x5938-0x593f	Send BD Ring 8 Host Producer Index
0x5940-0x5947	Send BD Ring 9 Host Producer Index
0x5948-0x594f	Send BD Ring 10 Host Producer Index
0x5950-0x5957	Send BD Ring 11 Host Producer Index
0x5958-0x595f	Send BD Ring 12 Host Producer Index
0x5960-0x5967	Send BD Ring 13 Host Producer Index
0x5968-0x596f	Send BD Ring 14 Host Producer Index
0x5970-0x5977	Send BD Ring 15 Host Producer Index
0x5978-0x597f	Send BD Ring 16 Host Producer Index
0x5980-0x5987	Send BD Ring 1 NIC Producer Index
0x5988-0x598f	Send BD Ring 2 NIC Producer Index
0x5990-0x5997	Send BD Ring 3 NIC Producer Index
0x5998-0x599f	Send BD Ring 4 NIC Producer Index
0x59a0-0x59a7	Send BD Ring 5 NIC Producer Index
0x59a8-0x59af	Send BD Ring 6 NIC Producer Index
0x59b0-0x59b7	Send BD Ring 7 NIC Producer Index

Table 415: Low-Priority Mailbox Registers (Cont.)

Offset	Registers
0x59b8-0x59bf	Send BD Ring 8 NIC Producer Index
0x59c0-0x59c7	Send BD Ring 9 NIC Producer Index
0x59c8-0x59cf	Send BD Ring 10 NIC Producer Index
0x59d0-0x59d7	Send BD Ring 11 NIC Producer Index
0x59d8-0x59df	Send BD Ring 12 NIC Producer Index
0x59e0-0x59e7	Send BD Ring 13 NIC Producer Index
0x59e8-0x59ef	Send BD Ring 14 NIC Producer Index
0x59f0-0x59f7	Send BD Ring 15 NIC Producer Index
0x59f8-0x59ff	Send BD Ring 16 NIC Producer Index

INTERRUPT MAILBOX 0 REGISTER (OFFSET 0X5800)

See "Interrupt Mailbox 0 Register (Offset 0x200)" on page 372.

OTHER INTERRUPT MAILBOX REGISTERS (OFFSET 0X5808-0X5818)

See "Other Interrupt Mailbox Registers (Offset 0x208-0x218)" on page 373.

GENERAL MAILBOX REGISTERS 1-8 (OFFSET 0X5820-0X5858)

See "General Mailbox Registers 1-8 (Offset 0x220-0x258)" on page 373.

RECEIVE BD STANDARD PRODUCER RING INDEX REGISTER (OFFSET 0X5868)

See "Receive BD Standard Producer Ring Index Register (Offset 0x268)" on page 373.

RECEIVE BD JUMBO PRODUCER RING INDEX REGISTER (OFFSET 0X5870)

See "Receive BD Jumbo Producer Ring Index Register (Offset 0x270)" on page 373.

RECEIVE BD MINI PRODUCER RING INDEX REGISTER (OFFSET 0X5878, BCM5700 AND BCM5701 ONLY)

See "Receive BD Mini Producer Ring Index Register (Offset 0x278)" on page 373.

RECEIVE BD RETURN RING 1-16 CONSUMER INDICES REGISTERS (OFFSET 0X5880-0X58F8)

See "Receive BD Return Ring 1-16 Consumer Indices Registers (Offset 0x280-0x2F8)" on page 373.

SEND BD RING 1-4 HOST PRODUCER INDICES REGISTERS (OFFSET 0X5900-0X5918)

See "Send BD Ring 1-16 Host Producer Indices Registers (Offset 0x300-0x378)" on page 374.

SEND BD RING 5-16 HOST PRODUCER INDICES REGISTERS (OFFSET 0X5920-0X5978, BCM5700 AND BCM5701 ONLY)

See "Send BD Ring 1-16 Host Producer Indices Registers (Offset 0x300-0x378)" on page 374.

SEND BD RING 1-16 NIC PRODUCER INDICES REGISTERS (OFFSET 0X5980-0X59F8, BCM5700 AND BCM5701 ONLY)

See "Send BD Ring 1-16 NIC Producer Indices Registers (Offset 0x380-0x3F8)" on page 374.

FLOW-THROUGH QUEUES

Table 416: Flow-Through Queues Registers

Offset	*Registers*
0x5C00-0xC403	FTQ Reset Register
0x5C04-0x5CB7	Reserved
0x5CB8-0x5CBB	MAC TX FIFO Enqueue Register
0x5CBC-0x5CC7	Reserved
0x5CC8-0x5CCB	RXMBUF Cluster Free Enqueue Register
0x5CCC-0x5CFB	Reserved
0x5CFC-0x5CFF	RDIQ FTQ Write/Peek Register

FTQ RESET REGISTER (OFFSET 0X5C00)

Table 417: FTQ Reset Register (Offset 0x5C00)

Bit	*Field*	*Descriptions*	*Init*	*Access*
31-18	Reserved	-	0	R/W
17	Reserved (BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	-	0	R/W
	Reset Type 2 Software Queue (other devices)	Set this bit to reset the Type 2 Software queue. When set to 0, this flow through queue is ready to use. This bit is self-clearing.		R/W
16	Reset Receive Data Completion FTQ	Set this bit to reset the Receive Data Completion flow through queue. When set to 0, this flow through queue is ready for use. This bit is self-clearing.		R/W
15	Reserved(BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	-	0	R/W
	Reset Receive Data and Receive BD Initiator FTQ (other devices)	Set this bit to reset the Receive Data and Receive BD Initiator flow through queue. When set to 0, this flow through queue is ready to use.		R/W
14	Reset Receive List Placement FTQ	Set this bit to reset the Receive List. This bit is self-clearing placement flow through queue. When set to 0, this flow through queue is ready to use. This bit is self-clearing.		R/W
13	Reset Receive BD Complete FTQ	Set this bit to reset the Receive BD Complete flow through queue. When set to 0, this flow through queue is ready for use. This bit is self-clearing.		R/W

Table 417: FTQ Reset Register (Offset 0x5C00) (Cont.)

Bit	Field	Descriptions	Init	Access
12	Reserved (BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	-	0	R/W
	Reset MBUF Cluster Free FTQ (other devices)	Set this bit to reset the MBUF Cluster Free flow through queue. When set to 0, this flow through queue is ready for use. This bit is self-clearing.		R/W
11	Reset MAC TX FTQ	Set this bit to reset the MAC TX flow through queue. When set to 0, this flow through queue is ready for use. This bit is self-clearing.		R/W
10	Reset Host Coalescing FTQ	Set this bit to reset the Host Coalescing flow through queue. When set to 0, this flow through queue is ready for use. This bit is self-clearing.		R/W
9	Reset Send Data Completion FTQ	Set this bit to reset the Send Data Completion flow through queue. When set to 0, this flow through queue is ready for use. This bit is self-clearing.		R/W
8	Reserved (BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	-	0	R/W
	Reset Type 1 Software FTQ (other devices)	Set this bit to reset the Type 1 Software flow through queue. When set to 0, this flow through queue is ready for use. This bit is self-clearing.		R/W
7	Reset DMA High Priority Write FTQ	Set this bit to reset the DMA High Priority Write flow through queue. When set to 0, this flow through queue is ready for use. This bit is self-clearing.		R/W
6	Reset DMA Write FTQ	Set this bit to reset the DMA Write flow through queue. When set to 0, this flow through queue is ready for use. This bit is self-clearing.		R/W
5	Reserved (BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	-	0	R/W
	Reset Send Data Initiator FTQ (other devices)	Set this bit to reset the Send Data Initiator flow through queue. When set to 0, this flow through queue is ready for use. This bit is self-clearing.		R/W
4	Reset Send BD Completion FTQ	Set this bit to reset the Send BD Completion flow through queue. When set to 0, this flow through queue is ready for use. This bit is self-clearing.		R/W
3	Reserved (BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	-	0	R/W
	Reset DMA Completion Discard FTQ (other devices)	Set this bit to reset the DMA Completion Discard flow through queue. When set to 0, this flow through queue is ready for use. This bit is self-clearing.		R/W

Table 417: FTQ Reset Register (Offset 0x5C00) (Cont.)

Bit	Field	Descriptions	Init	Access
2	Reset DMA High Priority Read FTQ	Set this bit to reset the DMA High Priority Read flow through queue. When set to 0, this flow through queue is ready for use. This bit is self-clearing.		R/W
1	Reset DMA Read Queue FTQ	Set this bit to reset the DMA Read Queue flow through queue. When set to 0, this flow through queue is ready for use. This bit is self-clearing.		R/W
0	Reserved	-		R/W

MAC TX FIFO ENQUEUE REGISTER (OFFSET 0X5CB8)

This register is applicable to BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only.

A write to this register will add a transmit packet to the tail of the MACTQ FTQ. The host CPU uses this register to send an ASF message out.

Since the size of TXMBUF FIFO is only 64 entries and MACTQ is 12 bits wide:

- Bits 21:16 from this register are mapped to bits 11:6 of the MACTQ FTQ.
- Bits 5:0 from this register are mapped to bits 5:0 of the MACTQ FTQ.
- Bits 31:22 and 15:6 are ignored.

The TXMBUF cluster for the ASF message is defaulted to the uppermost three TXMBUFs.

Table 418: MAC TX FIFO Enqueue Register (Offset 0x5CB8)

Bit	Field	Description	Init	Access
31-16	Head TXMBUF Pointer	Specifies the first MBUF of the TXMBUF cluster for the transmit packet.	0x003D	W/O
15-0	Tail TXMBUF Pointer	Specifies the last MBUF of the TXMBUF cluster for the transmit packet.	0x003F	W/O

RXMBUF CLUSTER FREE ENQUEUE REGISTER (OFFSET 0X5CC8)

This register is applicable to BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only.

A write to this register will free a cluster of RXMBUFs. The host CPU uses this register to deallocate RXMBUFs after it has processed the received ASF message.

Table 419: RXMBUF Cluster Free Enqueue Register (Offset 0x5CC8)

Bit	Field	Description	Init	Access
31-18	Reserved	-	0x0	W/O
17-9	Head RXMBUF Pointer	Specifies the first MBUF of the RXMBUF cluster for the received packet to be freed.	0x00	W/O
8-0	Tail RXMBUF Pointer	Specifies the last MBUF of the RXMBUF cluster for the received packet to be freed.	0x00	W/O

RDIQ FTQ WRITE/PEEK REGISTER (OFFSET 0X5CFC)

This register is applicable to BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only.

The host CPU uses this register to get the RXMBUF cluster pointers if the received packet requires the attention of the CPU. This could be an ASF or ACPI packet.

- A write to this register will modify the head of the RDIQ FTQ entry.
- A read of this register will peek at the head of the RDIQ FTQ entry.
- When the Valid bit is 1 and the Pass bit is 0, the CPU can take the RXMBUF cluster pointers to access the received packet (see Table 421).
- When the CPU writes a 1 to the Skip bit, the hardware will pop the head of the queue entry (see Table 421).

Table 420: RDIQ FTQ Write/Peek Register (Offset 0x5CFC)

Bit	Field	Description	Init	Access
31-21	Reserved	-	00000000000	R/O
20	Valid Bit	Set only if the head of the RDIQ entry is valid (i.e., the queue is non-empty). See Table 421.	0	R/W
19	Skip Bit	If this bit is set, the head of the RDIQ entry will be popped. The read pointer will be incremented. See Table 421.	0	R/W
18	Pass Bit	This bit is 0 if the RDIQ head entry is intended for the CPU. It prevents the entry to be serviced by WDMA. See Table 421.	0	R/W
17-9	Head RXMBUF Pointer	Specifies the first MBUF of the RXMBUF cluster for the received packet.	000000000	R/O
8-0	Tail RXMBUF Pointer	It specifies the last MBUF of the RXMBUF cluster for the received packet.	000000000	R/O

Table 421 shows the functional truth table for the combination of the Valid, Skip, and Pass bits.

Table 421: Functional Truth Table for the Combination of the Valid, Skip, and Pass Bits

Valid	Skip	Pass	Scenario
0	X	X	Head entry invalid.
1	0	0	Waiting for CPU to peek the head entry.
1	0	1	Waiting for WDMA to process the head entry.
1	1	0	CPU has finished peeking the head entry, the head of RDIQ will be popped in the next cycle.
1	1	1	Entry will be popped the next cycle. The WDMA might or might not have latched the head entry. (Not recommended).

MESSAGE SIGNALED INTERRUPT REGISTERS

Table 422: Message Signaled Registers

Offset	Registers
0x6000-0x6003	MSI Mode Register
0x6004-0x6007	MSI Status Register
0x6008-0x600b	MSI FIFO Access Register
0x600C-0x63FF	Reserved

MSI MODE REGISTER (OFFSET 0X6000)

Table 423: MSI Mode Register (Offset 0x6000)

Bit	Field	Description	Init	Access
31-30	Reserved (BCM5700 and BCM5701 only)	-	0	R/O
	Priority (other devices)	Sets the priority of the MSI engine relative to the DMA read engine and DMA Write engine. Equal settings result in fair round robin arbitration. • 00: Lowest • 01: Low • 10: High • 11: Highest	00	R/W
29-11	Reserved	-	0	R/O
10-8	MSI_MESSAGE (Applicable to BCM5752 Only)	This register sets the MSI message data bottom bits to the value programmed here. This register exists only for testing purposes and should always be programmed to zero.	000	R/W
	Reserved (other devices)	-	000	R/W
7	Reserved	-	0	R/O
6	MSI FIFO Overrun Attn (Not Applicable to BCM5752)	MSI FIFO overrun attention enable.	0	R/W
5	MSI FIFO Underrun Attn (Not Applicable to BCM5752)	MSI FIFO underrun attention enable.	0	R/W
4	PCI Parity Error Attn	PCI parity error attention enable.	0	R/W
3	PCI Master Abort Attn	PCI master abort attention enable.	0	R/W
2	PCI Target Abort Attn	PCI target abort attention enable.	0	R/W
1	Enable	This bit controls whether the MSI state machine is active or not. When set to 0, it completes the current operation and cleanly halts. Until it is completely halted, it remains one when read.	1	R/W
0	Reset	When this bit is set to 1, the MSI state machine is reset. This is a self-clearing bit.	0	R/W

MSI STATUS REGISTER (OFFSET 0X6004)

Table 424: MSI Status Register (Offset 0x6004)

Bit	Field	Description	Init	Access
31-7	Reserved	-	0	R/O
6	MSI FIFO Overrun (Not Applicable to BCM5752)	MSI FIFO overrun status	0	W2C
5	MSI FIFO Underrun (Not Applicable to BCM5752)	MSI FIFO underrun status	0	W2C
4	PCI Parity Error	PCI parity error status	0	W2C
3	PCI Master Abort	PCI master abort status	0	W2C
2	PCI Target Abort	PCI target abort status	0	W2C
1	Reserved	-	0	R/O
0	MSI_PCI_REQ (Applicable to BCM5752 only)	Reading this bit returns the current status of the request to PCI to send an MSI. If a value of 1 is read, then the request is currently asserted. Writing this bit with a value of one will cause the request to be asserted. Writing this bit with a value of 0 has no effect.	0	R/W
	Reserved (other devices)	-	0	R/O

MSI FIFO ACCESS REGISTER (OFFSET 0X6008)

This register is not applicable to BCM5752.

The MSI FIFO Access Register is used to give an MSI request to the PCI block. The actual MSI data is indicated in the bottom bits. If the MSI is properly enqueued into the FIFO, the overflow bit remains cleared. If the FIFO overflowed, the bit is set and must be written to be cleared. If the overflow bit is set when the access register is written, no MSI is enqueued.

Table 425: MSI FIFO Access Register (Offset 0x6008)

Bit	Field	Description	Init	Access
31-4	Reserved	Always 0.	0	R/O
3	Overflow	No space left in FIFO.	0	W2C
2-0	MSI Data	Indicates which of the (up to eight) MSIs to use.	0	W/O

DMA Completion Registers

Table 426: DMA Completion Registers

Offset	Registers
0x6400-0x6403	DMA Completion Mode Register
0x6404-0x67ff	Reserved

DMA Completion Mode Register (Offset 0x6400)

Table 427: MSI FIFO Access Register (Offset 0x6400)

Bit	Field	Description	Init	Access
31-2	Reserved	-	0	R/O
1	Enable	This bit controls whether the DMA Completion state machine is active or not. When set to 0, it completes the current operation and cleanly halts. Until it is completely halted, it remains one when read.		R/W
0	Reset	When this bit is set to 1, the DMA Completion state machine is reset. This is a self-clearing bit.		R/W

General Control Registers

Table 428: General Control Registers

Field	*Description*
0x6800-0x6803	Mode Control register
0x6804-0x6807	Misc Configuration register
0x6808-0x680b	Misc Local Control register
0x680c-0x680f	Timer register
0x6810-0x6813	RX-RISC Event register
0x6814-0x6817	RX-RISC Timer Reference register
0x6818-0x681b	RX-RISC Semaphore register
0x681c-0x681f	Remote RX-RISC Attention register
0x6820-0x6823	TX-RISC Event register
0x6824-0x6827	TX-RISC Timer Reference register
0x6828-0x682b	TX-RISC Semaphore register
0x682c-0x682f	Remote TX-RISC Attention register
0x6830-0x6837	Memory Power Up register
0x6838-0x683b	Serial EEPROM Address register
0x683c-0x683f	Serial EEPROM Data register
0x6840-0x6843	Serial EEPROM Control register
0x6844-0x6847	MDI Control register
0x6848-0x684b	Serial EEPROM Delay register
0x684c-0x684f	RX CPU Event Enable Register (offset 0x684C, BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)
0x6850-0x68ff	Reserved

MODE CONTROL REGISTER (OFFSET 0X6800)

Table 429: Mode Control Register (Offset 0x6800)

Bit	Field	Description	Init	Access
31	Reserved	-	0	R/O
30	Reserved (BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	-	0	R/O
	Route Multicast Frames to RISC Cores (other devices)	Forward multicast frames to RX and TX RISC processors. This bit should only be used with custom firmware.	0	R/W
29	Reserved (BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	-	0	R/O
	4x Size NIC-Based Send Rings (other devices)	Indicates that four NIC send rings are combined to create a single ring that are four times larger than normal (512 entries vs. 128 entries). Only valid if the host Send BDs bit is not set.	0	R/W
28	Interrupt on Flow Attention	Cause a host interrupt when an enabled flow attention occurs.	0	R/W
27	Interrupt on DMA Attention	Cause a host interrupt when an enabled DMA attention occurs.	0	R/W
26	Interrupt on MAC Attention	Cause a host interrupt when an enabled MAC attention occurs.	0	R/W
25	Interrupt on RX RISC Attention	Cause a host interrupt when an enabled RX RISC attention occurs.	0	R/W
24	Reserved (BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	-	0	R/O
	Interrupt on TX RISC Attention (other devices)	Cause a host interrupt when an enabled TX RISC attention occurs.	0	R/W
23	Receive No Pseudo-header Checksum	Do not include the pseudoheader in the TCP or UDP checksum calculations. To obtain the correct checksum, the driver must add the TCP/UDP checksum field to the pseudo-header checksum.	0	R/W
22	Reserved		0	R/O
21	NVRAM Write Enable (BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	The host must set this bit before attempting to update the Flash or SEEPROM (valid for BCM5703 B0 or later, BCM5704 A2 or later, BCM5705 A1 or later, BCM5721, and BCM5751. See "Revision Levels" on page 5.)	0	R/W
	Reserved (other devices)	-	0	R/O

Table 429: Mode Control Register (Offset 0x6800) (Cont.)

Bit	*Field*	*Description*	*Init*	*Access*
20	Send No Pseudo-header Checksum	Do not include the pseudoheader in the TCP or UDP checksum calculations. To obtain the correct checksum, the driver must seed the TCP/UDP checksum field with the pseudoheader checksum. ***Note:*** Some BCM57XX family adapters may calculate an incorrect pseudoheader checksum if this bit is left at the default value. Check the latest chip errata for affected chips and suggested workarounds.	0	R/W
19-18	Reserved	-	00	
17	Host Send BDs	Use host-based BD rings instead of NIC-based BD rings.	0	R/W
16	Host Stack Up	The host stack is ready to receive data from the NIC.	0	R/W
15	Force 32-bit PCI	Force PCI operation as if on a 32-bit PCI bus.	0	R/W
14	Don't Interrupt on Receives	Never cause an interrupt on receive return ring producer updates.	0	R/W
13	Don't Interrupt on Sends	Never cause an interrupt on send BD ring producer updates.	0	R/W
12	Reserved	-	0	
11	Allow Bad Frames	The RX MAC forwards illegal frames to the NIC and marks them as such instead of discarding them. The frames are queued based on default class and interrupt distribution queue number as specified in "Receive List Placement Configuration Register (Offset 0x2010)" on page 431).	0	R/W
10	Reserved	-	0	
9	No Frame Cracking	Turn off all frame cracking functionality in both the read DMA engine and the MAC receive engine. On receive, the TCP/UDP checksum field is replaced by raw checksum for the whole frame except the Ethernet header. On transmit, IP and TCP/UDP checksum generation is always disabled when this bit is set. Also, the raw checksum is calculated over the entire frame except the Ethernet header and CRC.	0	R/W
8	Reserved	-	0	R/O
7-6	GRC_Timeout_Cycles (BCM5714C/BCM5714S/ BCM5715C/BCM5715S A2 and later devices only)	• 00: GRC will timeout after 64k cycles • 01: GRC will timeout after 48k cycles • 10: GRC will timeout after 32k cycles • 11: GRC will timeout after 16k cycles	0	R/W
	Reserved (all other devices)	-	0	R/0
5	Word Swap Data	Word swap data when DMAing it across the PCI bus.	0	R/W
4	Byte Swap Data	Byte swap data when DMAing it across the PCI bus.	0	R/W
3	Reserved	-	0	R/O
2	Word Swap Non-frame Data	Word swap control structures (buffer descriptors, statistics) and data when DMAing them across the PCI bus.	0	R/W
1	Byte Swap Non-frame Data	Byte swap control structures (buffer descriptors, statistics) when DMAing them across the PCI bus.	0	R/W

Table 429: Mode Control Register (Offset 0x6800) (Cont.)

Bit	Field	Description	Init	Access
0	Reserved (BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	-	0	R/O
	Update on Coalescing Only (other devices)	When this bit is set, the NIC will only generate interrupts for transmit completions when the Send Coalesced Ticks value is exceeded. Send Consumer updates are still made based on the Send Max Coalesced BDs and the BD_FLAG_COAL_NOW.	0	R/W

Miscellaneous Configuration Register (Offset 0x6804)

The Miscellaneous Configuration register is used as an extension to the Miscellaneous Local Control register (see "Miscellaneous Local Control Register (Offset 0x6808)" on page 507). There are several fields used to control several small counters associated with the free-running 32-bit timer inside the device. The prescale function is performed on the clock prior to advancing the Timer register (see "Timer Register (Offset 0x680C)" on page 510) to provide a resolution as close as possible to 1 µs.

Table 430: Miscellaneous Configuration Register (Offset 0x6804)

Bit	Field	Description	Init	Access
31	Alternate Clock Enable (Applicable to only A2 and later versions of BCM5704 device)	The output of this bit is ORed with the Alternate Clock Control field of the PCI Clock Control register (see Table 179 on page 336). When this bit is: • Set, CORE CLK = 12.5 MHz and CPU_CLK = 25 MHz • Clear, CORE CLK = 66 MHz and CPU_CLK = 133 MHz	0 in full power mode, 1 in OOB mode	R/W
30	Reserved		0	R/O
29	Disable GRC Reset on PCIe Block (for BCM5721—A1 and later, BCM5751—A1 and later, BCM5714—A1 and later, and BCM5715—A1 and later only)	Setting this bit will prevent PCIe link training during a GRC reset.	0	R/W
29	Reserved (other devices)	-	0	R/O
28	Wire Speed Enable (BCM5721, BCM5751, and BCM5752 only)	When this bit is set, wire speed detection is enabled.	1	R/W
	ID_IN[5] (BCM5714 and BCM5715 only)	Status of ID bit 5.	0	R/O
	Reserved (other devices)	-	0	R/O
27	Wire Speed Timer Disable (BCM5721, BCM5751, and BCM5752 only)	When this bit is set, the wire speed timer is disabled.	0	R/W
	ID_IN[4] (BCM5714 and BCM5715 only)	Status of ID bit 4.	0	R/O
	Reserved 9other devices)	-	0	R/O

Table 430: Miscellaneous Configuration Register (Offset 0x6804) (Cont.)

Bit	Field	Description	Init	Access
26	GPHY Power Down Override (BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	When this bit is set, the GPHY will be left powered up when in the D0 uninitialized state. • In A1, this bit can only be cleared by a hard reset. A GRC or PCI reset has no effect. • In A0, this bit can be cleared by hard-reset, GRC reset, or PCI reset. ***Note:*** See "Revision Levels" on page 5.	0	R/W
	Reserved	-	0	R/O
25	DDQ_DLL Enable Disable (BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	When this bit is set, the handshake with the GPHY to power down the DLL is disabled. The IDDQ_DLL_Enable will always be 1.	0	R/W
	Revision ID (BCM5704 only)	When this bit is: Set, the BCM5704 is revision A2 or A3 Clear, the BCM5704 is revision A0, A1, or B0	X	R/O
	Reserved (other devices)	-	0	R/O
24	RAM Power Down (BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	When this bit is set, all of the RAMs are powered down.	0	R/W
	Reserved (other devices)	-	0	R/O
23	VREG Standby Current Mode (BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	When this bit is set, both vreg1 and vreg2 will be put into standby current mode (which consumes < 1 mA).	0	R/W
	ASF Strap (ServerWorks CIOB-E A1 or later only)	This bit reflects the value of the ASF strapping option on the ServerWorks CIOB-E, which has an integrated BCM5704. When this bit is: Set, the PCI function 0 controls the ASF interface Clear, the PCI function 1 controls the ASF interface	X	R/O
	Reserved (other devices)	-	0	R/O
22	BIAS IDDQ (BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	When this bit is set, the BIAS will be powered down.	0	R/W
	Module ID (BCM5704 A1 and later only)	This bit returns 0 for PCI function 0, and 1 for PCI function 1.	X	R/O
	Reserved (other devices)	-	0	R/O

Table 430: Miscellaneous Configuration Register (Offset 0x6804) (Cont.)

Bit	Field	Description	Init	Access
21	GPHY IDDQ (BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	When this bit is set, the GPHY will be powered down.	0	R/W
	PCI Power Present (BCM5704 A1 and later only)	When this bit is set, PCI power is present.	X	R/O
	Reserved (other devices)	-	0	R/O
20	Powerdown (BCM5704C and BCM5704S only)	Write 1 to power down the device. This bit is provided for the internal CPUs.	0	R/W
	Device Power Down (BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	Setting this bit will power down the device (power consumption is ~20 mW). This bit is cleared by PCI reset.	0	R/W
	Reserved (other devices)	-	0	R/O
19	VIO power state (BCM5702, BCM5703C, and BCM5703S only)	VIO is the PCI bus power	0	R/O
	PME EN State (BCM5704C and BCM5704S only)	State of PME Enable for this device.	1	R/O
	Reserved (other devices)	-	0	R/O
18-17	PowerState (BCM5704C and BCM5704S only)	Indicates the current power state of the device. • 00b: D0 • 01b: D1 • 02b: D2 • 03b: D3 This PowerState mirrors the PMSCR register (see "Power Management Control/Status Register (Offset 0x4C)" on page 318) power state bits and is read only to provide the internal CPUs with the current power state.	00	R/O
	Reserved	-	00	R/O
16–13	Bond ID (BCM5714, BCM5715, BCM5721, BCM5751, and BCM5752 Only)	For the: • BCM5714 device: - 0010b = BCM5714C - 0011b = BCM5714S • BCM5715 device: - 0110b = BCM5715C - 0111b = BCM5715S • BCM5751 device: - 0000b = BCM5751 - 0100b = BCM5751M • BCM5721 device, 0010b = BCM5721 • BCM5752 device: - 0000b = BCM5752 - 0100b = BCM5752M	ID(3:0)	R/O

Table 430: Miscellaneous Configuration Register (Offset 0x6804) (Cont.)

Bit	Field	Description	Init	Access
14	Bond ID 1 (BCM5704 only)	This bit reads as 1 for the ServerWorks CIOB-E, and 0 for the BCM5704.	X	R/O
13-8	Reserved	-	00	R/O
7-1	Timer Prescaler	Local Core clock frequency in MHz, minus 1, which should correspond to each advance of the timer. Reset to all 1. ***Example:*** A 66-MHz local core clock uses 65 (0x41).	1111111	R/W
0	CORE Clock Blocks Reset[1]	Write 1 to this bit resets the CORE_CLK blocks in the device. This is a self-clearing bit.	0	R/W

[1] PCIe configuration cycles are non-posted transactions and require a completion to avoid a PCIe bus error. Drivers using configuration cycles to do a GRC reset on 5751, 5721, and 5752 need to give the device enough time to send out the configuration write completion before the PCIe link goes down. The driver should slow the clock down by setting bits 20 (not 19) and 12 in register 0x74 before issuing configuration cycles for a GRC reset (bit 0 of register 0x6804) or setting the PWRDOWN bit (0x6804 bit 20). This is not required for the BCM5751, BCM5721, and BCM5752 devices with version A1 or later if PCIe reset is disabled during the GRC reset by setting the bit 29 of this register (0x6804) to 1.

Miscellaneous Local Control Register (Offset 0x6808)

The Miscellaneous Local Control register is used to control various functions within the device. All bits are set to zero (i.e. disabled) during reset.

Table 431: Miscellaneous Local Control Register (Offset 0x6808)

Bit	Field	Description	Init	Access
31	Enable Wolink Up (BCM5721 B0 and later, BCM5751 B0 and later, and BCM5752 only)	When set, the chip drives the PME when the link is up.	0	R/W
	GRC Alt Clock Enable (BCM5704, BCM5714, and BCM5715 only)	Write a 1 to this bit to enable alternate clock mode. This bit is used by firmware in V-AUX mode to put the chip into low-power mode.	0	R/W
	Reserved (Other devices)		0	R/O
30	Enable Wolink Down (BCM5721 B0 and later, BCM5751 B0 and later, and BCM5752 only)	When set, the chip drives the PME when the link is down.	0	R/W
	GRC Alt Clock Select (BCM5714 A1 & later and BCM5715 A1 & later only)	Write a 1 to this bit to increase the CPU/Core clock speed to normal speed. This bit is valid only in V-AUX mode (when 0x6804[19] = 0).	0	R/W
	Reserved (Other devices)		0	R/O
29	Disable Traffic LED Fix	Set to 1 to disable Traffic LED Fix	0	R/W
	Reserved	-	0	R/O
28	Select PCI Configuration (BCM5721, BCM5751, and BCM5752 only)	When this bit is set, the PCI mode will be controlled by bit 27.	0	R/W
	Ext_Ump_ID (BCM5714 and BCM5715 only)	This value is based on the strap option of PWR indicator pin.	0	R/O
	Reserved (other devices)	-	0	R/O

Table 431: Miscellaneous Local Control Register (Offset 0x6808) (Cont.)

Bit	Field	Description	Init	Access
27	Legacy PCI Mode Select	When this bit is set, the device will operate in PCI legacy mode.	0	R/W
	Int_Ump_ID (BCM5714 and BCM5715 only)	For NIC function 0, this bit will be 0 For NIC function 1, this bit will be 1.	0	R/O
26	Reserved (BCM5700 and BCM5701 only)	-	0	R/O
	PME Assert (all other devices)	When set, the PME Status bit in the PMSCR register (see "Power Management Control/Status Register (Offset 0x4C)" on page 318) is forced high. If PME Enable is also set, the PME signal will activate. This register bit is write-only and self-clearing after write.	0	R/O
25	Expansion ROM Code to MBUF3 (BCM5700, BCM5701, BCM5702, BCM5703, and BCM5704 only)	Set this bit to enable download of Expansion ROM into MBUF3. This bit must be reset if MBUF3 is used for packet buffering.	0	R/W
	Reserved (other devices)	-	0	R/O
24	Auto SEEPROM Access	If set, access to serial EEPROM goes through the serial EEPROM address and data registers. Otherwise, serial EEPROM control register should be used.	0	R/W
23	Reserved (all other devices)	-	0	R/O
	SSRAM Cycle Deselect (BCM5700 only)	If set, double cycles deselect is used. Otherwise, single cycle deselect is used.	0	R/W
22	SSRAM Type (BCM5700 only)	If set, ZBT SSRAM is in use. Otherwise, standard SSRAM is in use.	0	R/W
	Reserved (all other devices)		0	R/O
21	Bank Select (BCM5700 only)	If reset, two banks of SSRAM installed. Otherwise, only one bank of SSRAM is installed.	0	R/W
	Reserved (all other devices)		0	R/O
20-18	SRAM Size (BCM5700 only)	• 000 = 256 KB • 001 = 512 KB • 010 = 1 MB • 011 = 2 MB • 100 = 4 MB • 101 = 8 MB • 110 = 16 MB • 111 = Reserved	000	R/W
	Reserved (all other devices)	•	000	R/O
17	Enable External Memory (BCM5700 only)	Set to 1 if external memory is in use.	0	R/W
	Reserved (all other devices)		0	R/O
16-14	GPIO Pins [2:0] outputs	Outputs which are defined by board level design.	0	R/W
13-11	GPIO Pins [2:0] output enables	When asserted, the device drives miscellaneous pin outputs.	0	R/W
10-8	GPIO Pins [2:0] inputs	Input from bidirectional miscellaneous pin.	0	R/O

Table 431: Miscellaneous Local Control Register (Offset 0x6808) (Cont.)

Bit	*Field*	*Description*	*Init*	*Access*
7	Global Interrupt Enable (BCM5705, BCM5788, BCM5721, BCM5751, BCM5714, and BCM5715 only)	When this bit is set, the interrupt to the CPU is enabled.	0	R/W
	GPIO 3 Output (Applicable to BCM5752 only)	GPIO 3 Output Value	0	R/W
	Reserved (other devices)	-	0	R/O
6	GPIO 3 Output Enable (Applicable to BCM5752 only)	When set to 1, the GPIO 3 pin will be enabled as output pin.	0	R/W
	GPIO2_Gate_20uS (BCM5714C/BCM5714S/ BCM5715C/BCM5715S A2 and later devices only)	• 0: Behavior as in A1. • 1: GPIO2 output cannot drive low until PERST_B has been active for 20us.	0	R/W
	Reserved (other devices)		0	R/O
5	GPIO 3 Input Value (Applicable to BCM5752 only)	Input Value on GPIO 3	0	R/O
	SIGDET_Ctrl (BCM5714S/ BCM5715S A2 and later devices only)	• 0: SIGDET signal that feeds SERDES LINK is from internal SERDES block. • 1: SIGDET signal that feeds SERDES LINK is from external SIGDET pin	0	R/W
	Reserved (other devices)		0	R/O
4	LinkSignal_Gate (BCM5714S/BCM5715S A2 and later devices only)	• 0: Behavior as in A1 • 1: LINK signal is gated by SIGDET signal	0	R/W
	Reserved (other devices)		0	R/0
3	Interrupt on Attention	If set, the host will be interrupted when any of the attention bits in the CPU event register are asserted.	0	R/W
2	Set Interrupt	If Interrupt Mailbox 0 contains a nonzero value, setting this bit does nothing. If Interrupt Mailbox 0 is zero, then setting this bit will cause the internal unmasked interrupt state to be asserted. The external interrupt state (INTA pin) will also be asserted immediately if interrupts are not masked by the Mask Interrupts bit. If interrupts are masked, INTA will be asserted once interrupts are unmasked, so long as interrupts are not first cleared. This bit is not operational in MSI mode.	0	W/O
1	Clear Interrupt	This bit provides the same functionality as the Clear Interrupt bit in the Miscellaneous Host Control register. This bit is not operational in MSI mode	0	W/O
0	Interrupt State	This bit reflects the state of the PCI $\overline{\text{INTA}}$ pin. This bit is not operational in MSI mode.	0	R/O

Note: The GPIO pins of the BCM5704 are shared between both PCI functions of the device. Users writing their own drivers should ensure that the GPIO is not simultaneously enabled for input on one function and output on another function, as the results are unpredictable.

TIMER REGISTER (OFFSET 0X680C)

The Timer register is a 32-bit free-running counter. This counter increments when the Prescale Counter hits the Timer Prescaler limit as specified by the Miscellaneous Configuration register (see "Miscellaneous Configuration Register (Offset 0x6804)" on page 504). This counter is used by the CPU to keep track of relative time in microseconds. A write to the Timer register will load the counter value written.

Table 432: Timer Register (Offset 0x680C)

Bit	Field	Description	Init	Access
31-0	Timer value	32-bit free-running counter.	0	R/W

RX-RISC EVENT REGISTER (OFFSET 0X6810)

The RX-RISC uses the following event register. Software events are set by writing a one to the bit. The software event, timer event, and TX-RISC event are reset by writing a zero to the bit. Other events are based on hardware events and cannot be affected directly by the RISC processor.

Table 433: RX-RISC Event Register (Offset 0x6810)

Bit	Field	Description	Init	Access
31	SW Event 13	SW Event 13 is set.	0	R/W
30	SW Event 12	SW Event 12 is set.	0	R/W
29	Timer	Timer reference reached.	0	R/W
28	SW Event 11	SW Event 11 is set.	0	R/W
27	Flow Attn	Flow attention.	0	R/O
26	RX-CPU Attn	RX-RISC needs attention.	0	R/W
25	MAC Attn	MAC needs attention.	0	R/O
24	TX-CPU Attn	TX-RISC needs attention.	0	R/O
23	SW Event 10	SW Event 10 is set.	0	R/W
22	High-priority Mailbox	First 32 Mailbox registers have been updated.	0	R/O
21	Low-priority Mailbox	Last 32 Mailbox registers have been updated.	0	R/O
20	DMA Attn	A DMA channel needs attention.	0	R/O
19	SW Event 9	SW Event 9 is set.	0	R/W
18	High DMA RD (other devices) UMP_tx_rdv_evnt (BCM5714 and BCM5715 only)	High Priority DMA read FTQ has stalled. UMP receive block has frame to process.	0 0	R/O R/W
17	High DMA WR (all other devices) UMP_tx_rdv_evnt (BCM5714 and BCM5715 only)	High Priority DMA write FTQ has stalled. UMP transmit block can accept frame.	0 0	R/O R/W
16	SW Event 8	SW Event 8 is set.	0	R/W
15	Host Coalescing	The host coalescing FTQ has stalled.	0	R/O
14	SW Event 7	SW Event 7 is set.	0	R/W

Table 433: RX-RISC Event Register (Offset 0x6810) (Cont.)

Bit	Field	Description	Init	Access
13	Recv Data Comp (Post DMA)	Receive data completion FTQ has stalled.	0	R/O
12	SW Event 6	SW Event 6 is set.	0	R/W
11	RX SW Queue Event	Receive Software Queue Event.	0	R/W
10	DMA RD	Normal Priority DMA read FTQ has stalled.	0	R/O
9	DMA WR	Normal Priority DMA write FTQ has stalled.	0	R/O
8	Recv Data Init (Pre DMA)	Receive Data and Receive BD initiator FTQ has stalled.	0	R/O
7	SW Event 5	SW Event 5 is set.	0	R/W
6	Recv BD Comp	Receive BD completion FTQ has stalled.	0	R/O
5	SW Event 4	SW Event 4 is set.	0	R/W
4	Recv List Selector	Recv list selector is nonzero.	0	R/O
3	SW Event 3	SW Event 3 is set.	0	R/W
2	Recv List Placement (other devices)	Receive list placement FTQ has stalled.	0	R/O
	UMP_parity_evnt (BCM5714 and BCM5715 only)	UMP memory has detected parity error.	0	R/W
1	SW Event 1	SW Event 1 is set.	0	R/W
0	SW Event 0	SW Event 0 is set.	0	R/W

RX-RISC Timer Reference Register (Offset 0x6814)

The Timer Reference register allows the RX-RISC to receive an event when the free-running Timer register counts up to this value.

Table 434: RX-RISC Timer Reference Register (Offset 0x6814)

Bit	Field	Description	Init	Access
31-0	RX-CPU Timer Reference	RX-RISC Timer Event when Time stamp = RX-RISC Timer Reference. Reset to all 1.	0	R/W

RX-RISC Semaphore Register (Offset 0x6818)

The RX-RISC Semaphore register allows access to both internal RISC processors to a hardware semaphore mechanism. Writes to the register indicates the preference to toggle the own/not own states of a single semaphore bit. Reads of this register provide a 1 if that register owns the semaphore, and a 0 otherwise. To obtain the semaphore, the normal operation is a loop containing a write 0 followed by a read. Exit the loop when the read returns nonzero. To release the semaphore, the normal operation is to write 0.

Table 435: RX-RISC Semaphore Register (Offset 0x6818)

Bit	Field	Description	Init	Access
31-1	Reserved	-	0	R/O
0	RX-CPU Semaphore bit	-	0	R/W

Remote RX-RISC Attention Register (Offset 0x681C)

RX-RISC uses this register to set an event for the TX-RISC. Reading this register returns a zero.

Table 436: Remote RX-RISC Attention Register (Offset 0x681C)

Bit	Field	Description	Init	Access
31-1	Reserved	-	0	R/O
0	Set Attention	Writing a 1 to this bit sets the Remote RISC Attention bit in the TX-RISC Event register.	0	W/O

TX-RISC Event Register (Offset 0x6820)

These registers are only used on the BCM5700, BCM5701, BCM5702, BCM5703, and BCM5704. The TX-RISC uses the following event register. Software events are set by writing a one to the bit. The software event, timer event, and RX-RISC event are reset by writing a zero to the bit. Other events are based on hardware events and cannot be affected directly by the RISC processor.

Table 437: TX-RISC Event Register (Offset 0x6820)

Bit	Field	Description	Init	Access
31	SW Event 13	SW Event 13 is set.	0	R/W
30	SW Event 12	SW Event 12 is set R/W.	0	R/W
29	Timer	Timer reference reached.	0	R/W
28	SW Event 11	SW Event 11 is set.	0	R/W
27	Flow Attn	An unmasked bit is set in the Flow attention register.	0	R/O
26	TX-CPU Attn	TX-RISC needs attention.	0	R/W
25	MAC Attn	MAC needs attention.	0	R/O
24	RX-CPU Attn	RX-RISC needs attention.	0	R/O
23	SW Event 10	SW Event 10 is set.	0	R/W
22	High-priority Mailbox	First 32 Mailbox registers have been updated.	0	R/O
21	Low-priority Mailbox	Last 32 Mailbox registers have been updated.	0	R/O
20	DMA Attn	A DMA channel needs attention.	0	R/O
19	SW Event 9	SW Event 9 is set.	0	R/W
18	High DMA RD	High Priority DMA read FTQ has stalled.	0	R/O
17	High DMA WR	High Priority DMA write FTQ has stalled.	0	R/O
16	SW Event 8	SW Event 8 is set.	0	R/W
15	Host Coalescing	The host coalescing FTQ has stalled.	0	R/O
14	SW Event 7	SW Event 7 is set.	0	R/W
13	Send Data Comp (Post DMA)	Send data completion FTQ has stalled.	0	R/O
12	SW Event 6	SW Event 6 is set.	0	R/W
11	TX SW Queue Event	Transmit Software Queue Event.	0	R/W
10	DMA RD	Normal Priority DMA read FTQ has stalled.	0	R/O
9	DMA WR	Normal Priority DMA write FTQ has stalled.	0	R/O
8	Send Data Init (Pre DMA)	Send data initiator FTQ is set.	0	R/O

Table 437: TX-RISC Event Register (Offset 0x6820) (Cont.)

Bit	Field	Description	Init	Access
7	SW Event 5	SW Event 5 is set.	0	R/W
6	Send BD Comp	Send BD completion FTQ has stalled.	0	R/O
5	SW Event 4	SW Event 4 is set.	0	R/W
4	MAC TX	MAC TX FTQ has stalled.	0	R/O
3	SW Event 3	SW Event 3 is set.	0	R/W
2	SW Event 2	SW Event 2 is set.	0	R/W
1	SW Event 1	SW Event 1 is set.	0	R/W
0	SW Event 0	SW Event 0 is set.	0	R/W

TX-RISC Timer Reference Register (Offset 0x6824)

These registers are only used on the BCM5700, BCM5701, BCM5702, BCM5703, and BCM5704. The Timer Reference Register allows the TX-RISC to receive an event when the free-running Timer Register counts up to this value.

Table 438: TX-RISC Timer Reference Register (Offset 0x6824)

Bit	Field	Description	Init	Access
31-0	TX-CPU Timer Reference	TX-RISC Timer Event when Timer Register = TX-RISC Timer Reference. Reset to all 1.	0	R/W

TX-RISC Semaphore Register (Offset 0x6828)

These registers are only used on the BCM5700, BCM5701, BCM5702, BCM5703, and BCM5704. The TX-RISC Semaphore Register allow access to both internal RISC processors to a hardware semaphore mechanism. Writes to the register indicates the preference to toggle the own/not own states of a single semaphore bit. Reads of this register provide a 1 if that register owns the semaphore, and a 0 otherwise. To obtain the semaphore, the normal operation is a loop containing a write 0 followed by a read. Exit the loop when the read returns nonzero. To release the semaphore, the normal operation is to write 0.

Table 439: TX-RISC Semaphore Register (Offset 0x6828)

Bit	Field	Description	Init	Access
31-1	Reserved	-	0	R/O
0	Set Attention	-	0	R/W

Remote TX-RISC Attention Register (Offset 0x682C)

These registers are only used on the BCM5700, BCM5701, BCM5702, BCM5703, and BCM5704. TX-RISC uses this register to set an event for the RX-RISC. Reading this register returns zero.

Table 440: TX-RISC Attention Register (Offset 0x682C)

Bit	Field	Description	Init	Access
31-1	Reserved	-	0	R/O
0	Set Attention	Writing a 1 to this bit sets the TX-RISC Attention bit in the RX-RISC Event register.	0	R/W

Serial EEPROM Address Register (Offset 0x6838)

This 32-bit register is used by the RISCs in conjunction with the Serial EEPROM Data Register to read and/or write serial EEPROM data. The address register specifies the address and the direction of the transfer. When the transfer is complete (for either a read or a write), the complete bit is set.

To use this register pair to read the serial EEPROM, set the address and ensure the read/write bit is set in the address register. Loop reading the address register until the complete bit is set. When it is read the data from the data register. Clear the complete bit by writing the bit. No other transfer will occur when the complete bit is set. The Device ID must be programmed to select the appropriate device (A2 must be 0 for 128K/256Kx8 device).

To use this register pair to write the serial EEPROM, place the data into the data register. Then write the address into the address register ensuring that the write bit is clear. Loop reading the address register until the complete bit is set. When it is, the write is complete. Clear the complete bit by writing the bit. No other transfer will occur when the complete bit is set. It is the responsibility of software to control the timing between successive read/write access to the serial EEPROM.

Table 441: Serial EEPROM Address Register (Offset 0x6838)

Bit	Field	Description	Init	Access
31	Read/Write	If set, the transfer is a read.	0	R/W
30	Complete	Set when the transfer is complete.	0	W2C
29	Reset	Reset serial EEPROM hardware block.	0	R/W
28-26	Device ID	Device ID (A2, A1, A0).	0	R/W
25	Start Access	Trigger the hardware state machine to access the serial EE-PROM. This is a self-clearing bit.	0	R/W
24-16	Half Clock Period	Set the half clock period for the SEEPROM clock. Clock period = 2 x Half_Clock_Period x CORE_CLK.	0	R/W
15-2	SEEPROM address	Address of the word in SEEPROM to be read or written.	0	R/W
1-0	Reserved	Must be 0, byte addressing not available.	0	R/O

The software is responsible for handling the address rollover (page crossing) during the serial EEPROM access. For the read operations, only the Random Read Mode is supported.

SERIAL EEPROM DATA REGISTER (OFFSET 0X683C)

This 32-bit register holds the data to be written into the serial EEPROM or read from the serial EEPROM.

Table 442: Serial EEPROM Data Register (Offset 0x683C)

Bit	Field	Description	Init	Access
31-0	Data	Read/Write data register for the SEEPROM interface.	00000000h	R/W

SERIAL EEPROM CONTROL REGISTER (OFFSET 0X6840)

This serial EEPROM control register provides the CPU to toggle the pins to serial EEPROM directly. The Auto SEEPROM Access bit of the Miscellaneous Local Control register (see "Miscellaneous Local Control Register (Offset 0x6808)" on page 507) must be reset to enable the functions of this register.

Table 443: Serial EEPROM Control Register (Offset 0x6840)

Bit	Field	Description	Init	Access
31-6	Reserved	-	0	R/O
5	Data Input	Serial EEPROM data input	0	R/O
4	Data Output	Serial EEPROM data output control	0	R/W
3	Data Output Tri-state	Serial EEPROM data output tristate output control	0	R/W
2	Clock Input	Serial EEPROM clock input	0	R/O
1	Clock Output	Serial EERPOM clock output control	0	R/W
0	Clock Output Tri-state	Serial EEPROM clock output tristate control	0	R/W

MDI CONTROL REGISTER (OFFSET 0X6844)

The control register for handling the Management Data Interface, which used to communicate between the physical layer and management layer.

Table 444: MDI Control Register (Offset 0x6844)

Bit	Field	Description	Init	Access
31-4	Reserved	-	0	R/O
3	MDI Clock	When enabled, controls the clock signal at the MDC pin.	0	R/W
2	MDI Select	When set, the MDI interface is controlled by this register.	0	R/W
1	MDI Enable	When set, the MDI Data Pin is enabled as an output driver.	0	R/W
0	MDI Data	When read, returns the value at the MDIO pin. When written, and the MDI Enable bit is also set, the value is driven to the MDIO pin.	0	R/W

Serial EEPROM Delay Register (Offset 0x6848)

This 32-bit R/W register specifies the delay between the EEPROM access in 15 ns interval and is used for VPD access. Since the requirement of back-to-back write for Serial EEPROMs is 10ms, firmware currently programs this register to 0xA2C2A.

RX CPU Event Enable Register (Offset 0x684C)

This register is applicable to BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 devices only.

Setting a bit in this register enables an interrupt to the CPU or the event.

Table 445: RX CPU Event Enable Register (Offset 0x684C)

Bit	Field	Description	Init	Access
31	Flash	-	0	R/O
30	VPD	-	0	R/O
29	Timer Reference Reached	-	0	R/W
28	ROM	-	0	R/O
27	HC module	-	0	R/O
26	RX CPU module	-	0	R/O
25	EMAC module	-	0	R/O
24	Memory Map Enable Bit	Set by HW, cleared by SW.	0	R/W
23	Reserved	-	0	R/W
22	High-Priority Mail Box	-	0	R/O
21	Low-Priority Mail Box	-	0	R/O
20	DMA	-	0	R/O
19	Reserved	-	0	R/W
18-17	Reserved	-	00	R/W
16	ASF Location 15	-	0	R/W
15	TPM Interrupt Enable	-	0	R/W
14	ASF Location 14	-	0	R/W
13	Reserved	-	0	R/W
12	ASF Location 13	-	0	R/W
11	Unused SDI	-	0	R/W
10	SDC (Post TCP segmentation)	-	0	R/O
9	SDI (Pre TCP segmentation)	-	0	R/O
8	RDIQ FTQ (Received an ASF)	-	0	R/O
7	ASF Location 12	-	0	R/W
6	Reserved	-	0	R/W
5	ASF Location 11	-	0	R/W

Table 445: RX CPU Event Enable Register (Offset 0x684C) (Cont.)

Bit	Field	Description	Init	Access
4	Reserved	-	0	R/W
3	ASF Location 10	-	0	R/W
2	Reserved	-	0	R/W
1	ASF Location 9	-	0	R/W
0	ASF Location 8	-	0	R/W

Gig SerDes PRBS Control Register (0x6850, BCM5714 and BCM5715 Only)

Table 446: Gig SerDes PRBS Control Register (0x6850, BCM5714 only)

Bit	Field	Description	Init	Access
5	Prbs_soft_reset	Reset PRBS	0	R/W
4	Prbs_en	Enable PRBS	0	R/W
3	Prbs_inv	Invert PRBS pattern	0	R/W
2:1	Prbs_order	PRBS order	00	R/W
0	Prbs_error_clr	Clear PRBS error count	0	R/E

Gig SerDes PRBS Status Register (0x6854, BCM5714 and BCM5715 Only)

Table 447: Gig SerDes PRBS Status Register (0x6854, BCM5714 only)

Bit	Field	Description	Init	Access
15	Prbs_lock	PRBS monitor is locked	0	R
14	Prbs_lost_lock	PRBS lost lock (sticky bit)	0	R
13:0	Prbs_error_cnt	Error count (should be stable for pass)	0	R

GRC Message Exchange Out Register (0x6870H, BCM5714 and BCM5715 Only)

Table 448: GRC Message Exchange Out Register (0x6870H, BCM5714 only)

Bit	Field	Description	Init	Access
31:0	Message Out	Message Out to other function		RW

GRC Message Exchange In Register (0x6874H, BCM5714 and BCM5715 Only)

Table 449: GRC Message Exchange In Register (0x6874H, BCM5714 only)

Bit	Field	Description	Init	Access
31:0	Message In	Message In from other function		RW

WAKE-ON LAN REGISTERS

Note: The 0x6880 to 0x688B registers are only valid for the BCM5721, BCM5751, and BCM5752 devices.

Table 450: Wake-On-LAN Registers

Field	Description
0x6880-0x6883	WOL Mode Register
0x6884-0x6887	WOL Config Register
0x6888-0x688B	WOL State Machine Status Register

WOL MODE REGISTER (OFFSET 0X6880)

Table 451: WOL Mode Register (Offset 0x6880)

Bit	Field	Description	Init	Access
31-5	Reserved	-	0	R/W
4	WOL_PWR_CTRL	Use to manually switch the power FETs.	0	R/W
3-2	WOL_PWR_SW_PROG	Control the WOL voltage comparator threshold (100 mV increment).	0	R/W
1	Enable	This bit controls whether the WOL is active or not. • When set to 0, it completes the current operation and cleanly halts. • Until it is completely halted, it remains one when read.	0	R/W
0	Reset	When set, the entire WOL state machines are reset. This bit is self-clearing.	0	R/W

WOL Config Register (Offset 0x6884)

Table 452: WOL Config Register (Offset 0x6884)

Bit	Field	Description	Init	Access
31-17	Reserved	When set to 1, it indicates the main power is available.	0	R/W
16	SC_sim_mode	When set to 1, it accelerate the simulations. It waits only 0.5 µs. When set to 0, it is normal operation.	0	R/W
15-14	SC_delay_cfg	When set to 00, it accelerates the simulations for slow_clock_ctl. • 00: waits 10 µs. • 01: waits 35 µs. • 10: waits 50 µs. • 11: waits 65 µs.	0	R/W
13	SD_sim_mode	• 0: normal operation mode. The total delay is SD_delay_cfg x1000. • 1: waits 0.5 µs.	0	R/W
12-11	SD_delay_cfg	This is the amount of the delay. • 00: 0 • 01: 50 µs • 10: 150 µs • 11: 250 µs	0	R/W
10	Power_avail	When set to 1, it indicates the main power is available.	0	R/W
9	PHY_bypass	When set to 1, it indicates that it is emulation (IKOS) mode.	0	R/W
8	Lom_enable	When set to 1, LOM is enabled.	0	R/W
7-3	Reserved	-	0	R/W
2	WOL_done 2	When set to 1, WOL finishes its operations.	0	R/W
1	WOL_start 1	When set to 1, WOL state machines starts.	0	R/W
0	WOL_10	When set to 1, WOL is 10 Mbit only. When set to 0, WOL is 100 Mbit.	0	R/W

WOL State Machine Status Register (Offset 0x6888)

Table 453: WOL State Machine Status Register (Offset 0x6888)

Bit	Field	Description	Init	Access
31-29	Reserved	-	0	R/O
28-24	SD_state[4:0]	ShutDown control state machine value	0	R/O
23-20	Reserved	-	0	R/O
19-16	SC_state[3:0]	Slow_Clock control state machine value	0	R/O
15-13	Reserved	-	0	R/O
12-8	SW_state[4:0]	Setup_Wol control state machine value	0	R/O
7-3	Reserved	-	0	R/O
2-0	Wol_state[2:0]	WOL top control state machine value	0	R/O

MISCELLANEOUS TPM REGISTER

This register is applicable only to BCM5752 device.

MISCELLANEOUS TPM REGISTER (OFFSET 0X6890)

Table 454: Miscellaneous TPM Register (Offset 0x6890)

Bit	Field	Description	Init	Access
31-26	Misc R/W Reserved Bits	Reserved R/W bits that get reset by hard reset	0	R/W
25	Super Airplane Mode Enable	Read/write bit that controls whether super Airplane Mode is enabled.	0	R/W
24	Misc2 Bit	Reserved R/W bit that gets reset by power-on reset	0	R/W
23-1	Misc1 Bits	Reserved R/W bits that get reset by GRC reset	0	R/W
0	TPM IDDQ	This bit, when set, indicates that the TPM is in IDDQ mode.	0	R/O

FAST BOOT PROGRAM COUNTER REGISTER

This register is applicable only to BCM5752 device.

FAST BOOT PROGRAM COUNTER REGISTER (OFFSET 0X6894)

Table 455: Fast Boot Program Counter Register (Offset 0x6890)

Bit	Field	Description	Init	Access
31	FastBoot Enable	This bit is used by the CPU to keep track of whether or not there is valid phase 1 boot code stored in the RX MBUF. If the bit is set, then RXMBUF contains valid boot code. Otherwise, it is assumed that RXMBUF does not contain valid boot code. This bit is reset only by a power-on reset. The state of this bit has no effect on state machines within the device. It is used by the CPU to track boot code status.	0	R/W
30-0	FastBoot Program Counter	This field is used by the CPU to keep track of the location of the phase 1 boot code in RX MBUF. These bits behave identical to bit 31 in that they have no effect on state machine operation and they are cleared only by a power-on reset.	0	R/W

ASF SUPPORT REGISTERS (NOT APPLICABLE TO BCM5700)

Table 456: ASF Support Registers

Field	**Description**
0x6C00-0x6C03	ASF Control register
0x6C04-0x6C07	SMBus Input register
0x6C08-0x6C0b	SMBus Output register
0x6C0c-0x6C0f	Watchdog Timer
0x6C10-0x6C13	Heartbeat Timer
0x6C14-0x6C17	Poll ASF Timer
0x6C18-0x6C1b	Poll Legacy Timer
0x6C1c-0x6C1f	Retransmission Timer
0x6C20-0x6C23	Timestamp Counter
0x6C24-0x6C27	SMBus Driver Select register
0x6C28-0x6C2F	Reserved
0x6c30	TPM Command register
0x6c34	TPM Data register
0x6C38-0x6C3F	Reserved
0x6C40-0x6C43	Auxiliary SMBus Master Status register
0x6C44-0x6C47	Auxiliary SMBus Master Control register
0x6C48-0x6C4B	Auxiliary SMBus Master Command register
0x6C4C-0x6C4F	Auxiliary SMBus Block Data register
0x6C50-0x6C53	Auxiliary SMBus Slave Address/Control register
0x6C54-0x6C57	Auxiliary SMBus Slave Status register
0x6C58-0x6C5B	Auxiliary SMBus Slave Data register
0x6C5C-0x6C5F	Reserved
0x6c60-0x6c63	SMBus ARP Command register
0x6c64-0x6c67	SMBus ARP Status register
0x6c68-0x6c6b	UDID register 0
0x6c6c-0x6c6f	UDID register 1
0x6c70-0x6c73	UDID register 2
0x6c74-0x6c77	UDID register 3
0x6c80	Auxiliary SMBus Master Control Channel 1 register
0x6C84	Auxiliary SMBus Master Control Channel 1 register
0x6C88	Auxiliary SMBus Master Command Channel 1 register
0x6C8C	Auxiliary SMBus Block Data Channel 1 register
0x6c90	Auxiliary SMBus Slave Address/Control Channel 1 register
0x6c94	Auxiliary SMBus Slave Status Channel 1 register
0x6c98	Auxiliary SMBus Slave Data Channel 1 register

Table 456: ASF Support Registers (Cont.)

Field	Description
0x6cc0	Auxiliary SMBus Master Status Channel 2 register
0x6cc4	Auxiliary SMBus Master Control Channel 2 register
0x6cc8	Auxiliary SMBus Master Command Channel 2 register
0x6ccc	Auxiliary SMBus Block Data Channel 2 register
0x6cd0	Auxiliary SMBus Slave Address/Control Channel 2 register
0x6cd4	Auxiliary SMBus Slave Status Channel 2 register
0x6cd8	Auxiliary SMBus Slave Data Channel 2 register

ASF Control Register (Offset 0x6C00)

This register is not applicable to the BCM5700 MAC.

Table 457: ASF Control Register (Offset 0x6C00)

Bit	Field	Description	Init	Access
31	SMB Early Attn	When set, the SMB interface sets the ASF_GRC_ATTN bit as soon as slave activity is detected. When cleared, the attention bit is not set until an address match occurs.	0	R/W
30	SMB Enable ADDR 0	When set, the SMB interface accepts all incoming messages with an address of zero.	0	R/W
29-23	NIC SMB Address 2	Second NIC SMB address for matching incoming messages.	0	R/W
22-16	NIC SMB Address 1	First NIC SMB address for matching incoming messages.	0	R/W
15	SMB Autoread	When set, the SMB_IN_RDY bit in the SMB Input register (see "SMBus Input Register (Offset 0x6C04)" on page 524) will clear automatically whenever the register is read. Otherwise, the bit must be cleared by software.	0	R/W
14	SMB ADDR Filter	When clear, enables incoming SMBus message address filtering using the addresses specified in the NIC SMB Address 1 and NIC SMB Address 2 fields.	0	R/W
13	SMB Bit Bang Enable	When set, the SMBus bit-bang interface is enabled in the SMBus Output register.	0	R/W
12	SMB Enable	When set, the SMBus block is enabled.	0	R/W

Table 457: ASF Control Register (Offset 0x6C00) (Cont.)

Bit	*Field*	*Description*	*Init*	*Access*
11-8	ASF Attention Location	Controls which event bit in the Event Register the ASF Attention maps into. • 0 = Disabled • 1 = TXCPU event bit 1 • 2 = TXCPU event bit 2 • 3 = TXCPU event bit 3 • 4 = TXCPU event bit 5 • 5 = TXCPU event bit 7 • 6 = TXCPU event bit 12 • 7 = TXCPU event bit 14 • 8 = RXCPU event bit 0 • 9 = RXCPU event bit 1 • 10 = RXCPU event bit 3 • 11 = RXCPU event bit 5 • 12 = RXCPU event bit 7 • 13 = RXCPU event bit 12 • 14 = RXCPU event bit 14 • 15 = RXCPU event bit 16	0	R/W
7	SMB Attention	Set for incoming slave mode message.	0	W2C
6	Retransmission Timer Expired	Set when the retransmission timer has timed out.	0	W2C
5	Poll Legacy Timer Expired	Set when the Poll Legacy timer has timed out.	0	W2C
4	Poll ASF Timer Expired	Set when the Poll ASF timer has timed out.	0	W2C
3	Heartbeat Timer Expired	Set when the Heartbeat timer has timed out.	0	W2C
2	Watchdog Timer Expired	Set when the Watchdog timer has timed out.	0	W2C
1	Timestamp Counter Enable	Set to enable the time stamp counter.	0	R/W
0	ASF Reset	Soft reset bit for the ASF and SMBus interface blocks. When set, the blocks will be reset. The bit is self clearing.	0	R/W

Note: Some versions of the BCM57XX insert the SMBus Address byte from SMBus messages that should be filtered when the SMB ADDR filter is enabled. Broadcom recommends that customers writing their own SMBus interface routines not enable the SMB ADDR filter and perform SMBus address filtering in their software.

SMBus Input Register (Offset 0x6C04)

This register is not applicable to the BCM5700 MAC.

Table 458: SMBus Input Register (Offset 0x6C04)

Bit	Field	Description	Init	Access
31-14	Reserved.	Reserved for future use.	0	R/W
13-11	SMB Input Status	Value is set by the SMBus interface when the SMB Input Done bit is set. The value is encoded to the following: • 000: Reception OK. • 001: PEC error during reception. • 010: SMBus Input FIFO overflowed during reception. • 011: SMBus stopped unexpectedly during reception. • 100: SMBus timed out during reception.	000	R/W
10	SMBus In Firstbyte	Set by the SMBus interface block for the first byte received in the transfer.	0	R/W
9	SMBus In Done	Set by the SMBus block when the Data Input field has the last data byte of the transfer.	0	W2C
8	SMBus In Ready	Set by the SMBus interface block when the Data Input field is valid.	0	R/W
7-0	SMBus Data In	Input data from the SMBus interface.	0	R/W

Note: The BCM57XX uses a 5-byte internal input FIFO for SMBus messages that must be cleared if an error is indicated by the SMB Input Status field. This FIFO is cleared by continually reading the SMBus Data In field until the SMBus In Done bit is set, then clearing the SMBus In Done bit by writing a 1.

SMBus Output Register (Offset 0x6C08)

This register is not applicable to the BCM5700 MAC.

Table 459: SMBus Output Register (Offset 0x6C08)

Bit	Field	Description	Init	Access
31-29	Reserved.	Reserved for future use.	0	R/W
28	SMB Clock Input Value	Value on the SMB Clock pin when the SMBus interface is in bit-bang mode.	0	R/W
27	SMB Clock Enable	When set, the SMBus Clock signal is driven low when the SMBus interface bit-bang mode is also set. When clear, the SMBus Clock signal is tristated.	0	R/W
26	SMB Data Input Value	Value on the SMB Data pin when the SMBus interface is in bit-bang mode.	0	R/W
25	SMB Data Enable	When set, the SMBus Data signal is driven low when the SMBus interface bit-bang mode is also set. When clear, the SMBus Data signal is tri-stated.	0	R/W
24	SMB Slave Mode	Set when the SMBus interface is operating in slave mode.	0	R/W

Table 459: SMBus Output Register (Offset 0x6C08) (Cont.)

Bit	Field	Description	Init	Access
23-20	SMB Output Status	Set by SMBus interface when the SMB Output Start bit is cleared with the following encoded value that indicates the status of the preceding transfer: • 0000: Transmission OK. • 0001: SMBus was NACKed on the first byte of transmission. • 1001: SMBus was NACKed after the first byte of transmission. • 0010: SMBus Output FIFO underflowed during transmission. • 0011: SMBus stopped unexpectedly during transmission. • 0100: SMBus timed out during transmission. • 0101: SMBus Master lost arbitration during the first byte of transmission. • 1101: SMBus Master lost arbitration after the first byte of transmission. • 0110: Remote Master ACKed on what should have been the last byte.	0	R/W
19-14	SMB Read Length	Number of bytes in the read portion of the transaction.	0	R/W
13	Get Receive Length	When set, the receive length is taken from the first byte of the read data. When cleared, the SMB Read Length field is used.	0	R/W
12	Enable PEC	When set, the packet error check byte is enabled for the command.	0	R/W
11	SMB Access Type	When set, the SMBus interface will execute a read command. When cleared, the write command will be executed.	0	R/W
10	SMB Output Last	Set to indicate when the SMB Data Output field contains the last byte of the command.	0	R/W
9	SMB Output Start	Set to indicate the start of a SMBus master transaction. Cleared by the SMBus interface block when the transaction is complete.	0	R/W
8	SMB Output Ready	Set to indicate the SMB Data Output field has valid data. Cleared by the SMBus interface block when the bye is transferred to the internal FIFO.	0	R/W
7-0	SMB Data Output	Outgoing data byte for the SMB transaction.	0	R/W

Note: The BCM57XX uses a 5-byte internal output FIFO for SMBus messages. When an SMBus message is begun by setting the SMB Output Start bit, the software must write the next output byte within 100 µs, or an underflow may occur and invalidate the entire SMBus message.

ASF Watchdog Timer Register (Offset 0x6C0C)

This register is not applicable to the BCM5700 MAC.

Table 460: ASF Watchdog Timer Register (Offset 0x6C0C)

Bit	Field	Description	Init	Access
31-8	Reserved	Reserved for future use.	0	R/W
7-0	Watchdog timer	A countdown timer which decrements at the rate of one tick per second. When the counter reaches a value of zero, the corresponding timeout bit is set in the ASF Control Register (see "ASF Control Register (Offset 0x6C00)" on page 522). The timer stops decrementing when it reaches the zero value.	0	R/W

ASF Heartbeat Timer Register (Offset 0x6C10)

This register is not applicable to the BCM5700 MAC.

Table 461: ASF Heartbeat Timer Register (Offset 0x6C10)

Bit	Field	Description	Init	Access
31-16	Reserved.	Reserved for future use.	0	R/W
15-0	Heartbeat timer	A countdown timer which decrements at the rate of one tick per second. When the counter reaches a value of zero, the corresponding timeout bit is set in the ASF Control Register (see "ASF Control Register (Offset 0x6C00)" on page 522). The timer stops decrementing when it reaches the zero value.	0	R/W

Poll ASF Timer Register (Offset 0x6C14)

This register is not applicable to the BCM5700 MAC.

Table 462: Poll ASF Timer Register (Offset 0x6C14)

Bit	Field	Description	Init	Access
31-8	Reserved	Reserved for future use.	0	R/W
7-0	Poll timer	A countdown timer which decrements at the rate of one tick per 5 ms. When the counter reaches a value of zero, the corresponding timeout bit is set in the ASF Control Register (see "ASF Control Register (Offset 0x6C00)" on page 522). The timer stops decrementing when it reaches the zero value.	0	R/W

POLL LEGACY TIMER REGISTER (OFFSET 0X6C18)

This register is not applicable to the BCM5700 MAC.

Table 463: Poll Legacy Timer Register (Offset 0x6C18)

Bit	Field	Description	Init	Access
31-8	Reserved	Reserved for future use.	0	R/W
7-0	Poll Legacy timer	A countdown timer which decrements at the rate of one tick per 250 ms. When the counter reaches a value of zero, the corresponding timeout bit is set in the ASF Control Register (see "ASF Control Register (Offset 0x6C00)" on page 522). The timer stops decrementing when it reaches the zero value.	0	R/W

RETRANSMISSION TIMER REGISTER (OFFSET 0X6C1C)

This register is not applicable to the BCM5700 MAC.

Table 464: Retransmission Timer Register (Offset 0x6C1C)

Bit	Field	Description	Init	Access
31-8	Reserved	Reserved for future use.	0	R/W
7-0	Poll timer	A countdown timer which decrements at the rate of one tick per second. When the counter reaches a value of zero, the corresponding timeout bit is set in the ASF Control Register (see "ASF Control Register (Offset 0x6C00)" on page 522). The timer stops decrementing when it reaches the zero value.	0	R/W

TIME STAMP COUNTER REGISTER (OFFSET 0X6C20)

This register is not applicable to the BCM5700 MAC.

Table 465: Time Stamp Counter Register (Offset 0x6C20)

Bit	Field	Description	Init	Access
31- 0	Timestamp Counter	A count-up timer which increments at the rate of one tick per second. The counter starts when the Time Stamp Counter Enable bit is set in the ASF Control register (see "ASF Control Register (Offset 0x6C00)" on page 522).	0	R/W

SMBus Driver Select Register (Offset 0x6C24)

This version of the SMBus Driver Select register applies to the BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 MAC Transceivers only.

Table 466: SMBus Driver Select Register (Offset 0x6C24)

Bit	Field	Description	Init	Access
31- 1	Reserved	-		R/O
0	Driver Select	Set to 1 to enable SM_DATA_OUT and SM_CLK_OUT to use new SMBus interface.	0	R/W

Rest of BCM57XX Family

This version of the SMBus Driver Select register applies to the BCM5701 through BCM5704 MAC controllers.

Table 467: SMBus Driver Select Reg. (Offset 0x6C24, Rest of BCM57XX Fam. (Except BCM5700 MAC)

Bit	Field	Description	Init	Access
31- 0	Timestamp Counter	A count-up timer which increments at the rate of one tick per second. The counter starts when the Timestamp Counter Enable bit is set in the ASF Control register (see "ASF Control Register (Offset 0x6C00)" on page 522).	0	R/W

BCM5721, BCM5751, AND BCM5752 TPM INTERFACE REGISTERS

These registers are applicable to BCM5721, BCM5751, and BCM5752 only.

TPM COMMAND REGISTER (OFFSET 0X6C30)

This register is applicable to BCM5721, BCM5751, and BCM5752 only.

Table 468: TPM Command Register (Offset 0x6C30)

Bit	Field	Description	Init	Access
31-16	Register address	Register address value to TPM core. It is only valid when request is asserted. Firmware assigns this field based on TPM internal register mapping.	0x0	R/W
15-2	Reserved	-	0x0	R/W
1	Write/!Read	• 0: Read register access. • 1: Write register access. ***Note:*** WR/RD request will be asserted to the TPM core until ACK is given.	0	R/W
0	Start/!Done	Write 1 to start internal TPM register access. This bit is self-cleared to 0 once register access is completed and the read data (if read) is available in the Data register (see "TPM Data Register (Offset 0x6C34)" on page 529). The internal processor polls this bit to determine if the previous access is finished. ***Note:*** This bit is cleared by the tpm_ack signal from the TPM IP.	0	W/SC

TPM DATA REGISTER (OFFSET 0X6C34)

This register is applicable to BCM5721, BCM5751, and BCM5752 only.

Table 469: TPM Data Register (Offset 0x6C34)

Bit	Field	Description	Init	Access
31-0	Data value	Data to be written or return read data. Data valid only after Start/!Done bit is clear (see "TPM Command Register (Offset 0x6C30)" on page 529).	0x0	R/W

BCM5714 AND BCM5715 TPM INTERFACE REGISTERS

These registers are applicable to BCM5714 and BCM5715 only.

TPM COMMAND REGISTER (0X6C30)

Table 470: TPM Command Register (0x6c30, For BCM5714 and BCM5715 Only)

Bit	Field	Description	Init	Access
31-28	DB Reg Addr	Debug register address. SW should always write 0 to this field, other values are reserved for HW debug purpose.	0	R/W
27-8	Warm Ini	Warm-up timer initial value. Used by the random generator. SW should always write 0 to this field, other values will cause a premature Warm Done (bit 7) to be set.	0	R/W
7	Warm Done	The random generator initialization is complete. It normally (depends on the Warm Ini value) takes 2^21 core clocks from setting load Ini (bit 2) to complete the initialization.	0	R/O
6	Rnd Vld	The 32-bit random number in TPM Data Register (at OFFSET 0X6C34) is ready. Clear this bit by writing 1 to Upd Rnd (bit 5).	0	R/O
5	Upd Rnd	Update the 32-bit random data. Writing 1 to this bit triggers the random generator to start generating a new 32-bit random number.	0	W2C
4	Attn Msk	Attention Mask. Writing 1 to this bit disables the attention signaling when Rnd Vld (bit 6) is set.	0	R/W
3	Div2	Divide. When configured low, the random number generation is twice as fast as high.	0	R/W
2	Load Ini	Load Warm up timer initial value. Writing 1 to this bit clears the Warm Done (bit 7) and the internal initialization timer starts incrementing based on the Warm Ini (bit 27-8) value until the Warm Done is set.	0	W2C
1	Rnd En	Random generator enable.	0	R/W
0	Rnd Rst	Random generator reset.	0	W2C

TPM DATA REGISTER (0X6C34)

Table 471: TPM Data Register (0x6C34)

Bit	Field	Description	Init	Access
31-0	Rnd32bit	32-bit random number.	0	R

AUXILIARY SMBUS MASTER STATUS REGISTER (OFFSET 0X6C40)

This register is for the BCM5704, BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only.

Table 472: Auxiliary SMBus Master Status Register (Offset 0x6C40)

Bit	Field	Description	Init	Access
31-8	Reserved	-	0	R/O
7	CRC/PEC Error	• 0: No CRC/PEC Error detected. • 1: CRC/PEC Error Detected. This bit is set only by hardware and can be reset by writing a 1 to this position. ***Note:*** Was bit-5 on 5705 by mistake	0	R/W
6	Reserved	-	0	R/O
5	CRC/PEC Error for BCM5705. (Please see bit-7). Reserved for all other devices	-	0	R/O
4	Failed	• 0: SMBus Attention not caused by KILL bit. • 1: Source of the SMBus attention is a failed bus transaction, set when KILL bit in SMB Master Control register is set. This bit is set only by hardware and can be reset by writing a one to this position.	0	R/W
3	Bus Collision	• 0: SMBus Attention not caused by transaction collision. • 1: Source of SMBus Attention was a transaction collision. This bit is set only by hardware and can be reset by writing a 1 to this position.	0	R/W
2	Device Error	• 0: SMBus interrupt not caused by transaction error. • 1: Source of SMBus interrupt was the generation of a SMBus transaction error. This bit is set only by hardware and can be reset by writing a 1 to this position. Transaction errors are usually caused by: Illegal command field, Unclaimed Cycle, Master Device Time-out.	0	R/W
1	SMBus Attention	• 0: SMBus attention not caused by Master command completion. • 1: Source of SMBus attention was the completion of the last Master command. This bit is set only by hardware and can be reset by writing a 1 to this position.	0	R/W
0	Master Busy	• 0: SMBus Controller Master interface is not processing a command. • 1: Indicates that the SMBus controller master interface is in the process of completing a command. None of the other SMBus Master registers should be accessed if this bit is set.	0	R/O

Auxiliary SMBus Master Control Register (offset 0x6c44)

This register is applicable to BCM5704, BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 devices only.

Table 473: Auxiliary SMBus Master Control Register (Offset 0x6C44)

Bit	Field	Description	Init	Access
31-30	Reserved	-	0	R/O
29	SMBus Slave Soft Reset (BCM5714, BCM5721 and BCM5751 only)	Setting this bit will reset the SMBus slave interface.	0	R/W
	Reserved	-	0	R/O
28	SMBus Softreset	• 0: Normal operation. • 1: Resets the SMBus Interface.	0	R/W
27-24	SM Module Attention Selection (BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	This field selects which GRC Attention is asserted when an SMBus interrupt occurs.	0	R/W
	Reserved (other devices)	-	0	R/O
23-20	Reserved	-	0	R/O
19	SM_DATA Pin Control	• 0: Drive the SM_DATA pin low. • 1: No functional impact on the SM_DATA pin.	0	R/W
18	SM_CLK Pin Control	• 0: Drive the SM_CLK pin low. • 1: No functional impact on the SM_CLK pin.	0	R/W
17	SM_DATA Pin Current Status	This bit returns the value on the SM_DATA pin. This allows software to read the current state of the pin.	0	R/O
16	SM_CLK Pin Current Status	This bit returns the value on the SM_CLK pin. This allows software to read the current state of the pin.	0	R/O
15-11	Reserved	-	0	R/O
10	Slave Read Attention Enable	• 0: SMBus Slave will not wait for status bit to be cleared before supplying the data. • 1: SMBus Slave will stretch the clock until the status bit is cleared.	0	R/W
9	Bit-Bang Interface Enable	• 0: Bit-Bang Interface Disabled. • 1: Bit-Bang Interface Enabled.	0	R/W
8	SM Bus Speed	• 0: 100 Hz SMBus Interface. • 1: 400 Hz SMBus Interface.	0	R/W
7	CRC/PEC Enable	• 0: Disable CRC/PEC. • 1: Enable CRC/PEC generation.	0	R/W
6	Start	• 0: Has no effect. Always read a 0. • 1: Execution start. Writing a 1 in this field initiates SMBus controller Master Interface to execute the command programmed in the SMB command port field.	0	R/W
5	Reserved	-	0	R/O

Table 473: Auxiliary SMBus Master Control Register (Offset 0x6C44) (Cont.)

Bit	Field	Description	Init	Access
4-2	SMBus Command Protocol	Select the type of command the SMBus Controller Master interface will execute. Reads and writes are determined by bit 0 of SMBus master address register: • 000: Quick Read or Write. • 001: Byte Read or Write. • 010: Byte Data Read or Write. • 011: Word Data Read or Write. • 100: Reserved. • 101: Block Read or Write. • 110: Block write-block read process call. • 111: Reserved.	000	R/W
1	Kill	• 0: This will allow the Master Controller interface function to continue normally. • 1: Stop the current Master transaction in process. This sets the failed status bit and asserts interrupt selected by the SMB interrupt select field.	0	R/W
0	Interrupt Enable	• 0: Disable the generation of Attention • 1: Enable the generation of Attention on the completion of current Master transaction.	0	R/W

AUXILIARY SMBUS MASTER COMMAND REGISTER (OFFSET 0X6C48)

This register is applicable to BCM5704, BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 devices only.

Table 474: Auxiliary SMBus Master Command Register (Offset 0x6C48)

Bit	Field	Description	Init	Access
31-24	SMBus Data 1	This register should be programmed with a value to be transmitted in the data 1 field of an SMBus Master Interface Word Transaction.	0x00	R/W
23-16	SMBus Data 0	This register should be programmed with a value to be transmitted in the data 0 field of an SMBus Master Interface Word Transaction. • For Block write commands, the count of the memory should be stored in this field. The value of this register is loaded into the block transfer count field. This register must be set to a value between 1 and 32 for block command counts. • For block reads, count received from SMBus device is stored here.	0x00	R/W
15-9	SMBus Address	This field contains the 7-bit address of the targeted slave device.	0	R/W
8	SMBus Read or Write	• 0: Execute a Write Command. • 1: Execute a Read Command.	0	R/W
7-0	SMBus Master Command	This field contains the data transmitted in the command field of SMBus Master transaction.	0x00	R/W

AUXILIARY SMBUS BLOCK DATA REGISTER (OFFSET 0X6C4C)

This register is applicable to BCM5704, BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 devices only.

Table 475: Auxiliary SMBus Block Data Register (Offset 0x6C4C)

Bit	Field	Description	Init	Access
31-8	Reserved	-	0	R/W
7-0	SMBus Block Data	This register is used to transfer data into or out of the block data storage array. For Block read and Write commands.	0x00	R/W

AUXILIARY SMBUS SLAVE ADDRESS/CONTROL REGISTER (OFFSET 0X6C50)

This register is applicable to BCM5704, BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 devices only.

Table 476: Auxiliary SMBus Slave Address/Control Register (Offset 0x6C50)

Bit	Field	Description	Init	Access
31-8	Reserved	-	0	R/W
7-1	SMBus Slave Address	Only meaningful if AV flag is set. User also needs to program bit 0 of register 0x6C64 AV_REG to mark address valid based on SMBus 2.0 spec.	0	R/W
0	Slave Enable	• 0: Disable • 1: Enable Slave Interface. (This bit must also be set for ARP offload on BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715.)	0	R/W

AUXILIARY SMBUS SLAVE STATUS REGISTER (OFFSET 0X6C54)

This register is applicable to BCM5704, BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 devices only.

Table 477: Auxiliary SMBus Slave Status Register (Offset 0x6C54)

Bit	Field	Description	Init	Access
31-3	Reserved	-	0	R/W
2	Slave Read Requested	• 0: SMBus Attention not caused by slave. • 1: Source of SMBus attention is slave read cycle that matched the SMB Slave address. This bit is only set by hardware and can be reset by writing a 1 to this position. Slave interface stretches the clock until this bit is cleared. Read request for ARP will not trigger this bit. ARP hardware will supply read data in wire speed.	0	R/W
1	Slave Cycle Complete	• 0: SMBus Attention not caused by slave. • 1: Source of SMBus attention is completion of a slave cycle that matched the SMB Slave address. This bit is only set by hardware and can be reset by writing a 1 to this position. Completion for ARP will not trigger this bit. ARP Status register contains that information.	0	R/W
0	Slave Busy	• 0: SMBus Controller slave interface is not processing data. • 1: Indicates that the SMBus Controller slave interface is in the process of receiving data. None of the other SMBus Slave registers should be accessed if this bit is set. **Note:** This bit is also set during ARP process.	0	R/O

Auxiliary SMBus Slave Data Register (Offset 0x6C58)

This register is applicable to BCM5704, BCM5705, BCM5788, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 devices only.

Table 478: Auxiliary SMBus Slave Data Register (Offset 0x6C58)

Bit	Field	Description	Init	Access
31-22	Reserved	-	0	R/W
21-16	Write Byte Count (BCM5704, BCM5705, and BCM5788 only)	Indicates the number of bytes written into the FIFO by the SMBus master.	0	R/W
	Read Byte Count (BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	Indicates the number of bytes read from the FIFO by the Host.	0	R/W
15-14	Reserved	-	0	R/W
13-8	Read Byte Count (BCM5704, BCM5705, and BCM5788 only)	Indicates the number of bytes read from the FIFO by the Host.	0	R/W
	Write Byte Count (BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	Indicates the number of bytes written into the FIFO by the SMBus master.	0	R/W
7-0	Slave Data	-	0	R/O

Take the following software precautions:

- The software should set receive accelerate mode bit (see bit 10 of the "Write DMA Mode Register (Offset 0x4C00)" on page 480) only when the PCI bus speed is 33 MHz and the Core clock speed is 62.5 MHz.
- The software should set the long burst mode (see bits 17-16 of the "Read DMA Mode Register (Offset 0x4800)" on page 477) only when the PCI bus speed is 33 MHz and the Core clock speed is 62.5 MHz.
- The software should set the Bus-Parking save mode (see bit 23 of the "PCI Clock Control Register (Offset 0x74)" on page 334) to 1 when the PCI bus speed is 66 MHz.
- When in the RDMA long burst mode (i.e., bits 17-16 = 11 of the "Read DMA Mode Register (Offset 0x4800)" on page 477), the difference between the producer index and consumer index should be less than or equal to 63. This boundary condition is needed to prevent PCI burst reading of more than 2K bytes because PCI module only looks at bit[10:0] of the request length from either DMA engines. This restriction is removed in A3.
- The software should clear bit[22] of the DMA Read/Write Control Register (see the "DMA Read/Write Control Register (Offset 0x6C)" on page 327) when entering slow core clock mode. Otherwise, the BCM5705 could run into a danger of asserting $\overline{\text{REQ}}$ for more than 16 PCI clock cycles without issuing an active $\overline{\text{FRAME}}$ on the bus.

SMBus Address Resolution Protocol Registers (Offset 0x6CE0)

This register is applicable to BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 devices only

Table 479: SMBus ARP Command Register (Offset 0x6CE0)

Bit	Field	Description	Init	Access
31:17	Reserved	-	0x0	R/W
16	ARP attention enable	• 0: No attention is asserted to the internal CPU after the ARP command is completed. • 1: Attention is asserted to the internal CPU after the ARP command is completed. Attention stays until the CPU clears the attention source by clearing command completed bit.	0	R/W
15:14	Reserved	-	00	R/W
13	Directed Reset enable	• 0: Disable Directed Reset command capability. • 1: Enable Directed Reset command capability.	0	R/W
12	Directed Get UDID enable	• 0: Disable Directed Get UDID command capability. • 1: Enable Directed Get UDID command capability.	0	R/W
11	Assign Address enable	• 0: Disable Assign Address command capability. • 1: Enable Assign Address command capability.	0	R/W
10	General Get UDID enable	• 0: Disable General Get UDID command capability. • 1: Enable General Get UDID command capability.	0	R/W
9	General Reset Device enable	• 0: Disable General Reset Device command capability. • 1: Enable General Reset Device command capability.	0	R/W
8	Prepare to ARP enable	• 0: Disable Prepare to ARP command capability. • 1: Enable Prepare to ARP command capability.	0	R/W
7:3	Reserved	-	00000	R/W
2	ARP software reset	• 0: Normal operation. • 1: Reset ARP state machine.	0	R/W
1	PSA enable	• 0: Not Persistent Slave Address enabled. • 1: Persistent Slave Address enabled.	0	R/W
0	ARP enable	• 0: Disable ARP HW offload. • 1: Enable ARP HW offload. **Note:** SMBus Slave interface has to be enabled in addition to this bit for ARP to process.	0	R/W

SMBus ARP Status Register (Offset0x6CE4)

This register is applicable to BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 devices only

Table 480: SMBus ARP Status Register (Offset 0x6CE4)

Bit	Field	Description	Init	Access
31:16	Reserved	-	0	
15:14	Reserved	-	00	
13	Directed Reset completion	• 0: Has not received a valid Direct Reset command. • 1: Has received a valid Direct Reset command. This bit is set by hardware and a write 1 to clear. This is independent of the interrupt enable bit.	0	R/W2C
12	Directed Get UDID completion	• 0: Has not received a valid Direct Get UDID command. • 1: Has received a valid Direct Get UDID command. This bit is set by hardware and a write 1 to clear. This is independent of the interrupt enable bit.	0	R/W2C
11	Assign Address completion	• 0: Has not received a valid Assign Address command. • 1: Has received a valid Assign Address command. This bit is set by hardware and a write 1 to clear. This is independent of the interrupt enable bit.	0	R/W2C
10	General Get UDID completion	• 0: Has not received a valid General Get UDID command. • 1: Has received a valid General Get UDID command. This bit is set by hardware and a write 1 to clear. This is independent of the interrupt enable bit.	0	R/W2C
9	General Reset Device completion	• 0: Has not received a valid General Reset Device command. • 1: Has received a valid General Reset Device command. This bit is set by hardware and a write 1 to clear. This is independent of the interrupt enable bit.	0	R/W2C
8	Prepare to ARP completion	• 0: Has not received a valid Prepare to ARP command. • 1: Has received a valid Prepare to ARP command. This bit is set by hardware and a write 1 to clear. This is independent of the interrupt enable bit.	0	R/W2C
7:3	Reserved	-	00000	
2	ARP Busy	• 0: ARP HW is in idle state. • 1: APR HW is in process of ARP messages. This bit is only set by hardware.	0	R/O
1	AR: Address Resolved Flag	• 0: Current Slave Address is not resolved based on the ARP. • 1: Current Slave Address is resolved based on the ARP.	0	R/W
0	AV: Address Valid Flag	• 0: Current Slave Address is not valid. • 1: Current Slave Address is valid.	0	R/W

UDID REGISTER 0 (OFFSET 0X6CE8)

This register is applicable to BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 devices only

Table 481: UDID Register 0 (Offset 0x6CE8)

Bit	Field	Description	Init	Access
31:0	Vendor Specific ID	A unique number per device	0x0	R/W

UDID REGISTER 1 (OFFSET 0X6CEC)

This register is applicable to BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 devices only

Table 482: UDID Register 1 (Offset 0x6CEC)

Bit	Field	Description	Init	Access
31:16	Subsystem Vendor ID	This field may hold a value derived from any of several resources. ***Example:*** As assigned by PCI SIG.	0x0	R/W
15:0	Subsystem Device ID	Identifies a specific interface, implementation, or device.	0x0	R/W

UDID REGISTER 2 (OFFSET 0X6CF0)

This register is applicable to BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 devices only

Table 483: UDID Register 2 (Offset 0x6CF0)

Bit	Field	Description	Init	Access
31:16	Device ID	The device ID assigned by the device manufacturer.	0x0	R/W
15:0	Interface	Identifies the protocol layer interfaces supported over the SMBus connection by the device. ***Example:*** ASF	0x0	R/W

UDID REGISTER 3 (OFFSET 0X6CF4)

This register is applicable to BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 devices only

Table 484: UDID Register 3 (Offset 0x6CF4)

Bit	Field	Description	Init	Access
	Device Capability	Describes the device's capabilities.	0x0	R/W
	Version/Revision	The UDID version number, and a silicon revision identification.	0x0	R/W
	Vendor ID	The device manufacturer's ID as assigned by the SBS implementers' Forum or PCI SIG.	0x0	R/W

AUXILIARY SMBUS MASTER STATUS CHANNEL 1 REGISTER (OFFSET 0X6C80)

This register is applicable only to the BCM5704 MAC Transceiver.

Table 485: Auxiliary SMBus Master Status Channel 1 Register (Offset 0x6C80, BCM5704 Only)

Bit	Field	Description	Init	Access
31-8	Reserved	-	0	R/O
7	CRC/PEC Error	• 0: No CRC/PEC Error detected. • 1: CRC/PEC Error Detected. This bit is only set by hardware and can be reset by writing a 1 to this position.	0	R/W
	Reserved	-	0	R/O
6:5	Reserved	-	0	R/O
4	Failed	• 0: SMBus Attention not caused by KILL bit. • 1: Source of the SMBus attention is a failed bus transaction, set when KILL bit in SMB Master Control register is set. This bit is only set by hardware and can be reset by writing a one to this position.	0	R/W
3	Bus Collision	• 0: SMBus Attention not caused by transaction collision. • 1: Source of SMBus Attention was a transaction collision.This bit is only set by hardware and can be reset by writing a 1 to this position.	0	R/W
2	Device Error	• 0: SMBus interrupt not caused by transaction error. • 1: Source of SMBus interrupt was the generation of a SMBus transaction error. This bit is only set by hardware and can be reset by writing a 1 to this position. Transaction errors are usually caused by: Illegal command field, Unclaimed Cycle, Master Device Time-out.	0	R/W
1	SMBus Attention	• 0: SMBus attention not caused by Master command completion. • 1: Source of SMBus attention was the completion of the last Master command. This bit is only set by hardware and can be reset by writing a 1 to this position.	0	R/W
0	Master Busy	• 0: SMBus Controller Master interface is not processing a command. • 1: Indicates that the SMBus controller master interface is in the process of completing a command. None of the other SMBus Master registers should be accessed if this bit is set.	0	R/O

AUXILIARY SMBUS MASTER CONTROL CHANNEL 1 REGISTER (OFFSET 0X6C84)

This register is applicable only to the BCM5704 MAC Transceiver.

Table 486: Auxiliary SMBus Master Control Channel 1 Register (Offset 0x6C84, BCM5704 Only)

Bit	*Field*	*Description*	*Init*	*Access*
31-29	Reserved	-	0	R/O
28	SMBus Softreset	• 0: Normal operation • 1: Resets the SMBus Interface	0	R/W
27-20	Reserved	-	0	R/O
19	SM_DATA Pin Control	• 0: Drive the SM_DATA pin low • 1: No functional impact on the SM_DATA pin	0	R/W
18	SM_CLK Pin Control	• 0: Drive the SM_CLK pin low • 1: No functional impact on the SM_CLK pin	0	R/W
17	SM_DATA Pin Current Status	This bit returns the value on the SM_DATA pin. This allows software to read the current state of the pin.	0	R/O
16	SM_CLK Pin Current Status	This bit returns the value on the SM_CLK pin. This allows software to read the current state of the pin.	0	R/O
15-11	Reserved	-	0	R/O
10	Slave Read Attention Enable	• 0: SMBus Slave will not wait for status bit to be cleared before supplying the data. • 1: SMBus Slave will stretch the clock until the status bit is cleared.	0	R/W
9	Bit-Bang Interface Enable	• 0: Bit-Bang Interface Disabled • 1: Bit-Bang Interface Enabled	0	R/W
8	SM Bus Speed	• 0: 100 Hz SMBus Interface • 1: 400 Hz SMBus Interface	0	R/W
7	CRC/PEC Enable	• 0: Disable CRC/PEC • 1: Enable CRC/PEC generation	0	R/W
6	Start	• 0: Has no effect. Always read a 0. • 1: Execution start. Writing a 1 in this field initiates SMBus controller Master Interface to execute the command programmed in the SMB command port field.	0	R/W
5	Reserved	-	0	R/O
4-2	SMBus Command Protocol	Select the type of command the SMBus Controller Master interface will execute. Reads and writes are determined by bit 0 of SMBus master address register: • 000: Quick Read or Write • 001: Byte Read or Write • 010: Byte Data Read or Write • 011: Word Data Read or Write • 100: Reserved • 101: Block Read or Write • 110: Block write-block read process call • 111: Reserved	000	R/W
1	Kill	• 0: This will allow the Master Controller interface function to continue normally. • 1: Stop the current Master transaction in process. This sets the failed status bit and asserts interrupt selected by the SMB interrupt select field.	0	R/W

Table 486: Auxiliary SMBus Master Control Channel 1 Register (Offset 0x6C84, BCM5704 Only) (Cont.)

Bit	Field	Description	Init	Access
0	Interrupt Enable	• 0: Disable the generation of Attention. • 1: Enable the generation of Attention on the completion of current Master transaction.	0	R/W

AUXILIARY SMBUS MASTER COMMAND CHANNEL 1 REGISTER (OFFSET 0X6C88)

This register is applicable only to the BCM5704 MAC Transceiver.

Table 487: Auxiliary SMBus Master Command Channel 1 Register (Offset 0x6C88, BCM5704 Only)

Bit	Field	Description	Init	Access
31-24	SMBus Data 1	This register should be programmed with a value to be transmitted in the data 1 field of an SMBus Master Interface Word Transaction.	0x00	R/W
23-16	SMBus Data 0	This register should be programmed with a value to be transmitted in the data 0 field of an SMBus Master Interface Word Transaction. • For block write commands, the count of the memory should be stored in this field. The value of this register is loaded into the block transfer count field. This register must be set to a value between 1 and 32 for block command counts. • For block reads, count received from SMBus device is stored here.	0x00	R/W
15-9	SMBus Address	This field contains the 7-bit address of the targeted slave device.	0	R/W
8	SMBus Read or Write	• 0: Execute a Write Command • 1: Execute a Read Command	0	R/W
7-0	SMBus Master Command	This field contains the data transmitted in the command field of SMBus Master transaction.	0x00	R/W

AUXILIARY SMBUS BLOCK DATA CHANNEL 1 REGISTER (OFFSET 0X6C8C)

This register is applicable only to the BCM5704 MAC Transceiver.

Table 488: Auxiliary SMBus Block Data Channel 1 Register (Offset 0x6C8C, BCM5704 Only)

Bit	Field	Description	Init	Access
31-8	Reserved	-	0	R/W
7-0	SMBus Block Data	This register is used to transfer data into or out of the block data storage array. For Block Read and Write commands.	0x00	R/W

AUXILIARY SMBUS SLAVE ADDRESS/CONTROL CHANNEL 1 REGISTER (OFFSET 0X6C90)

This register is applicable only to the BCM5704 MAC Transceiver.

Table 489: Auxiliary SMBus Slave Address/Control Channel 1 Register (Offset 0x6C90, BCM5704 Only)

Bit	Field	Description	Init	Access
31-8	Reserved	-	0	R/W
7-1	SMBus Slave Address	-	0	R/W
0	Slave Enable	• 0: Disable. • 1: Enable Slave Interface.	0	R/W

AUXILIARY SMBUS SLAVE STATUS CHANNEL 1 REGISTER (OFFSET 0X6C94)

This register is applicable only to the BCM5704 MAC Transceiver.

Table 490: Auxiliary SMBus Slave Status Channel 1 Register (Offset 0x6C94, BCM5704 Only)

Bit	Field	Description	Init	Access
31-3	Reserved	-	0	R/W
2	Slave Read Requested	• 0: SMBus Attention not caused by slave. • 1: Source of SMBus attention is slave read cycle that matched the SMB Slave address. This bit is only set by hardware and can be reset by writing a 1 to this position. Slave interface stretches the clock until this bit is cleared.	0	R/W
1	Slave Cycle Complete	• 0: SMBus Attention not caused by slave. • 1: Source of SMBus attention is completion of a slave cycle that matched the SMB Slave address. This bit is only set by hardware and can be reset by writing a 1 to this position.	0	R/W
0	Slave Busy	• 0: SMBus Controller slave interface is not processing data. • 1: Indicates that the SMBus Controller slave interface is in the process of receiving data. None of the other SMBus Slave registers should be accessed if this bit is set.	0	R/O

Auxiliary SMBus Slave Data Channel 1 Register (Offset 0x6C98)

This register is applicable only to the BCM5704 MAC Transceiver.

Table 491: Auxiliary SMBus Slave Data Channel 1 Register (Offset 0x6C98, BCM5704 Only)

Bit	Field	Description	Init	Access
31-22	Reserved	-	0	R/W
21-16	Write Byte Count	Indicates the number of bytes written into the FIFO by the SMBus master.	0	R/W
15-14	Reserved	-	0	R/W
13-8	Read Byte Count	Indicates the number of bytes read from the FIFO by the Host.	0	R/W
7-0	Slave Data	-	0	R/O

Auxiliary SMBus Master Status Channel 2 Register (Offset 0x6CC0)

This register is applicable only to the BCM5704 MAC Transceiver.

Table 492: Auxiliary SMBus Master Status Channel 2 Register (Offset 0x6CC0, BCM5704 Only)

Bit	Field	Description	Init	Access
31-8	Reserved	-	0	R/O
7	CRC/PEC Error	• 0: No CRC/PEC Error detected. • 1: CRC/PEC Error Detected. This bit is only set by hardware and can be reset by writing a 1 to this position.	0	R/W
	Reserved	-	0	R/O
6:5	Reserved	-	0	R/O
4	Failed	• 0: SMBus Attention not caused by KILL bit. • 1: Source of the SMBus attention is a failed bus transaction, set when KILL bit in SMB Master Control register is set. This bit is only set by hardware and can be reset by writing a one to this position.	0	R/W
3	Bus Collision	• 0: SMBus Attention not caused by transaction collision. • 1: Source of SMBus Attention was a transaction collision.This bit is only set by hardware and can be reset by writing a 1 to this position.	0	R/W
2	Device Error	• 0: SMBus interrupt not caused by transaction error. • 1: Source of SMBus interrupt was the generation of a SMBus transaction error. This bit is only set by hardware and can be reset by writing a 1 to this position. Transaction errors are usually caused by Illegal command field, Unclaimed Cycle, or Master Device Time-out.	0	R/W

Table 492: Auxiliary SMBus Master Status Channel 2 Register (Offset 0x6CC0, BCM5704 Only) (Cont.)

Bit	Field	Description	Init	Access
1	SMBus Attention	• 0: SMBus attention not caused by Master command completion. • 1: Source of SMBus attention was the completion of the last Master command. This bit is only set by hardware and can be reset by writing a 1 to this position.	0	R/W
0	Master Busy	• 0: SMBus Controller Master interface is not processing a command. • 1: Indicates that the SMBus controller master interface is in the process of completing a command. None of the other SMBus Master registers should be accessed if this bit is set.	0	R/O

AUXILIARY SMBUS MASTER CONTROL CHANNEL 2 REGISTER (OFFSET 0X6CC4)

This register is applicable only to the BCM5704 MAC Transceiver.

Table 493: Auxiliary SMBus Master Control Channel 2 Register (Offset 0x6CC4, BCM5704 Only)

Bit	Field	Description	Init	Access
31-29	Reserved	-	0	R/O
28	SMBus Softreset	• 0: Normal operation. • 1: Resets the SMBus Interface.	0	R/W
27-20	Reserved	-	0	R/O
19	SM_DATA Pin Control	• 0: Drive the SM_DATA pin low. • 1: No functional impact on the SM_DATA pin.	0	R/W
18	SM_CLK Pin Control	• 0: Drive the SM_CLK pin low. • 1: No functional impact on the SM_CLK pin.	0	R/W
17	SM_DATA Pin Current Status	This bit returns the value on the SM_DATA pin. This allows software to read the current state of the pin.	0	R/O
16	SM_CLK Pin Current Status	This bit returns the value on the SM_CLK pin. This allows software to read the current state of the pin.	0	R/O
15-11	Reserved	-	0	R/O
10	Slave Read Attention Enable	• 0: SMBus Slave will not wait for status bit to be cleared before supplying the data. • 1: SMBus Slave will stretch the clock until the status bit is cleared	0	R/W
9	Bit-Bang Interface Enable	• 0: Bit-Bang Interface Disabled. • 1: Bit-Bang Interface Enabled.	0	R/W
8	SM Bus Speed	• 0: 100 Hz SMBus Interface. • 1: 400 Hz SMBus Interface.	0	R/W
7	CRC/PEC Enable	• 0: Disable CRC/PEC. • 1: Enable CRC/PEC generation.	0	R/W
6	Start	• 0: Has no effect. Always read a 0. • 1: Execution start. Writing a 1 in this field initiates SMBus controller Master Interface to execute the command programmed in the SMB command port field	0	R/W
5	Reserved	-	0	R/O

Table 493: Auxiliary SMBus Master Control Channel 2 Register (Offset 0x6CC4, BCM5704 Only) (Cont.)

Bit	Field	Description	Init	Access
4-2	SMBus Command Protocol	Select the type of command the SMBus Controller Master interface will execute. Reads and writes are determined by bit 0 of SMBus master address register: • 000: Quick Read or Write • 001: Byte Read or Write • 010: Byte Data Read or Write • 011: Word Data Read or Write • 100: Reserved • 101: Block Read or Write • 110: Block write-block read process call • 111: Reserved	000	R/W
1	Kill	• 0: This will allow the Master Controller interface function to continue normally. • 1: Stop the current Master transaction in process. This sets the failed status bit and asserts interrupt selected by the SMB interrupt select field.	0	R/W
0	Interrupt Enable	• 0: Disable the generation of Attention. • 1: Enable the generation of Attention on the completion of current Master transaction.	0	R/W

AUXILIARY SMBUS MASTER COMMAND CHANNEL 2 REGISTER (OFFSET 0X6CC8)

This register is applicable only to the BCM5704 MAC Transceiver.

Table 494: Auxiliary SMBus Master Command Channel 2 Register (Offset 0x6CC8, BCM5704 Only)

Bit	Field	Description	Init	Access
31-24	SMBus Data 1	This register should be programmed with a value to be transmitted in the data 1 field of an SMBus Master Interface Word Transaction.	0x00	R/W
23-16	SMBus Data 0	This register should be programmed with a value to be transmitted in the data 0 field of an SMBus Master Interface Word Transaction. • For Block write commands, the count of the memory should be stored in this field. The value of this register is loaded into the block transfer count field. This register must be set to a value between 1 and 32 for block command counts. • For block reads, count received from SMBus device is stored here.	0x00	R/W
15-9	SMBus Address	This field contains the 7-bit address of the targeted slave device.	0	R/W
8	SMBus Read or Write	• 0: Execute a Write Command. • 1: Execute a Read Command.	0	R/W
7-0	SMBus Master Command	This field contains the data transmitted in the command field of SMBus Master transaction.	0x00	R/W

AUXILIARY SMBUS BLOCK DATA CHANNEL 2 REGISTER (OFFSET 0X6CCC)

This register is applicable only to the BCM5704 MAC Transceiver.

Table 495: Auxiliary SMBus Block Data Channel 2 Register (Offset 0x6CCC, BCM5704 Only)

Bit	Field	Description	Init	Access
31-8	Reserved	-	0	R/W
7-0	SMBus Block Data	This register is used to transfer data into or out of the block data storage array. For Block read and Write commands.	0x00	R/W

AUXILIARY SMBUS SLAVE ADDRESS/CONTROL CHANNEL 2 REGISTER (OFFSET 0X6CD0)

This register is applicable only to the BCM5704 MAC Transceiver.

Table 496: Auxiliary SMBus Slave Address/Control Channel 2 Register (Offset 0x6CD0, BCM5704 Only)

Bit	Field	Description	Init	Access
31-8	Reserved	-	0	R/W
7-1	SMBus Slave Address	-	0	R/W
0	Slave Enable	• 0: Disable. • 1: Enable Slave Interface.	0	R/W

AUXILIARY SMBUS SLAVE STATUS CHANNEL 2 REGISTER (OFFSET 0X6CD4)

This register is applicable only to the BCM5704 MAC Transceiver.

Table 497: Auxiliary SMBus Slave Status Channel 2 Register (Offset 0x6CD4, BCM5704 Only)

Bit	Field	Description	Init	Access
31-3	Reserved	-	0	R/W
2	Slave Read Requested	• 0: SMBus Attention not caused by slave. • 1: Source of SMBus attention is slave read cycle that matched the SMB Slave address. This bit is only set by hardware and can be reset by writing a 1 to this position. Slave interface stretches the clock until this bit is cleared.	0	R/W
1	Slave Cycle Complete	• 0: SMBus Attention not caused by slave. • 1: Source of SMBus attention is completion of a slave cycle that matched the SMB Slave address. This bit is only set by hardware and can be reset by writing a 1 to this position.	0	R/W
0	Slave Busy	• 0: SMBus Controller slave interface is not processing data. • 1: Indicates that the SMBus Controller slave interface is in the process of receiving data. None of the other SMBus Slave registers should be accessed if this bit is set.	0	R/O

AUXILIARY SMBUS SLAVE DATA CHANNEL 2 REGISTER (OFFSET 0X6CD8)

This register is applicable only to the BCM5704 MAC Transceiver.

Table 498: Auxiliary SMBus Slave Data Channel 2 Register (Offset 0x6CD8, BCM5704 Only)

Bit	Field	Description	Init	Access
31-22	Reserved	-	0	R/W
21-16	Write Byte Count	Indicates the number of bytes written into the FIFO by the SMBus master.	0	R/W
15-14	Reserved	-	0	R/W
13-8	Read Byte Count	Indicates the number of bytes read from the FIFO by the Host.	0	R/W
7-0	Slave Data	-	0	R/O

NON-VOLATILE MEMORY INTERFACE REGISTERS (NOT APPLICABLE TO BCM5700 OR BCM5701)

Table 499: Non-Volatile Memory Interface Registers

Address	*Description*
0x7000	NVM Command Register
0x7004	Reserved
0x7008	NVM Write Register
0x700c	NVM Address Register
0x7010	NVM Read Register
0x7014	NVM Config 1 Register
0x7018	NVM Config 2 Register
0x701c	NVM Config 3 Register
0x7020	Software Arbitration Register
0x7024	NVM Access Register
0x7028	NVM Write1 Register
0x702c	NVM Arbitration Watchdog Timer Register

NVM COMMAND REGISTER (OFFSET 0X7000)

This register is not applicable to the BCM5700 and BCM5701 MAC Transceivers.

Table 500: NVM Command Register (Offset 0x7000)

Bit	Field	Description	Init	Access
31-28	Policy Error Reserved (other devices)	Reports Address Lockout Policy Error violations.	0	R/O
28-20	Reserved			
19	wrsr (BCM5714 and BCM5715 only)	The write status register command bit. Setting to 1 makes the flash interface state machine generate wrsr_cmd(0x1, hard-wired) to the flash device through the SPI interface to set the status register of the flash device to be written with sr_data. For SST25VF512 only.	0	R/W
18	ewsr (BCM5714 and BCM5715 only)	The enable write status register command bit. Setting to 1 makes the flash interface state machine generate ewsr_cmd(0x50, hard-wired) to the flash device through the SPI interface to set the status register of the flash device to be write-enabled. For SST25VF512 only.	0	R/W
17	wrdi (BCM5714 and BCM5715 only)	The write disable command bit. Setting to 1 makes the flash interface state machine generate wrdi_cmd (see 0x7028) to the flash device through the SPI interface to set the flash device to be write-disabled. Used for the device with protection function.		R/W
16	wren (BCM5714 and BCM5715 only)	The write enable command bit. Setting to 1 makes the flash interface state machine generate wren_cmd (see 0x7028) to the flash device through the SPI interface to set the flash device to be write-enabled. Used for the device with protection function.	0	R/W
15-9	Reserved	-	0	R/O
8	last	When this bit is set, the next command sequence is interpreted as the last one of a burst and any cleanup work is done. This means that the buffer is written to flash memory if needed on a write.	0	R/W
7	first	This bit is passed to the SEE_FSM or SPI_FSM if the pass_mode bit is set.	0	R/W
6	erase	The erase command bit. Set high to execute an erase. This bit is ignored if the wr is clear.	0	R/W
5	wr	The Write/Not_Read command bit. Set to execute write or erase.	0	R/W
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.	0	R/W
3	done	Sequence completion bit that is asserted when the command requested by assertion of the doit bit has completed. The done bit will be cleared while the command is in progress. The done bit 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.	0	WTC
2-1	Reserved	-	0	R/O
0	Reset	When set, the entire NVM state machine is reset. This bit is self clearing.	0	R/W

NVM STATUS REGISTER (0X7004H)

Table 501: NVM Status Register (0x7004H)

Bit	Field	Description	Init	Access
31-13	Reserved	-		
7-4	SEE_FSM State	State Machine Values (TBD)	0	R/O
7-4	SEE_FSM State	State Machine Values (TBD)	0	R/O
3-0	SPI_FSM State	State Machine Values (TBD)	0	R/O

NVM WRITE REGISTER (OFFSET 0X7008)

This register is not applicable to the BCM5700 and BCM5701 MAC Transceivers.

Table 502: NVM Write Register (Offset 0x7008)

Bit	Field	Description	Init	Access
31-0	wrdata (BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	32 bits of write data are used when write commands are executed. When bitbang_mode is set, bits 0 to 3 control the drive value of the SCK, CS_L, SO, and SI pins respectively.	0	R/W
	wrdata (BCM5703, BCM5704, and BCM5705 only)	32 bits of write data are used when write commands are executed. When bitbang_mode is set, bits 0 to 5 control the drive value of the SCL, SDA, SCK, CS_L, SO, and SI pins respectively. When pass_mode is set bits 7:0 will be the data to be written to EEPROM or Flash device.	0	R/W

NVM ADDRESS REGISTER (OFFSET 0X700C)

This register is not applicable to the BCM5700 and BCM5701 MAC Transceivers.

Table 503: NVM Address Register (Offset 0x700C)

Bit	Field	Description	Init	Access
31-24	Reserved	-	0	R/O
23-0	wraddr (BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	24-bit address value. When bitbang_mode is set, bits 0 to 3 control the OE value of the SCK, CS_L, SO, and SI pins respectively. ***Note:*** The OE of SCL, SDA, and SI is active high and the OE of SCK, CS_L, and SO is active low.	0	R/W
	wraddr (BCM5703, BCM5704, and BCM5705 only)	24-bit address value. When bitbang_mode is set, bits 0 to 5 control the OE value of the SCL, SDA, SCK, CS_L, SO, and SI pins respectively.	0	R/W

NVM READ REGISTER (OFFSET 0X7010)

This register is not applicable to the BCM5700 and BCM5701 MAC Transceivers.

Table 504: NVM Read Register (Offset 0x7010)

Bit	Field	Description	Init	Access
31-0	rddata (BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	32 bits of read data are used when read commands are executed. When bitbang_mode is set, bits 0 to 3 reflect the current input value of the SCK, CS_L, SO, and SI pins respectively.	0	R/W (BCM5752 only) R/O (all other devices)
	rddata (BCM5703, BCM5704, and BCM5705 only)	32 bits of read data are used when read commands are executed. When bitbang_mode is set, bits 0 to 5 reflect the current input value of the SCL, SDA, SCK, CS_L, SO, and SI pins respectively. When pass_mode is set, bits 7:0 will be the data read from EEPROM or Flash device.	0	R/W

NVM CONFIG 1 REGISTER (OFFSET 0X7014)

This register is not applicable to the BCM5700 and BCM5701 MAC Transceivers.

Table 505: NVM Config 1 Register (Offset 0x7014)

Bit	Field	Description	Init	Access
31	compat_bypass	Enable the 5701 legacy SEEPROM interface to bypass this interface.	0	R/W
30-28	PageSize (BCNM5752 only)	These bits indicate the page size of the attached flash device. They are set automatically depending on the chosen flash as indicated by the strapping option pins. Page sizes are as follows: • 000b = 256 bytes • 001b = 512 bytes • 010b = 1024 bytes • 011b = 2048 bytes • 100b = 4096 bytes • 101b = 264 bytes • 110b = Reserved • 111b = Reserved	Depends on Flash Strapping	R/W
27	Address Lockout Enable Status (BCM5752 only)	This bit will be set if the address lockout feature is active; it will be clear otherwise. This bit is read only. Its state can be changed only via a strapping option or bonding option.	Depends on Address Lockout Enable State	R/O
	Reserved (other devices)		0	R/O

Table 505: NVM Config 1 Register (Offset 0x7014) (Cont.)

Bit	Field	Description	Init	Access
26	Safe Erase (BCM5752 only)	When mimicking buffered behavior with an unbuffered Flash, this bit controls whether a page erase occurs as the result of a write first or a write last. If the bit is clear, then the erase will be issued as part of the write first operation. If the bit is set, then the erase shall be issued the first step in a write last operation. Clearing the bit may result in higher performance, as the erase occurs concurrent with the write more operations. Setting the bit may result in improved data integrity, as the erase is not started until the flash controller is ready to write back the entire page.	1	R/W
	Reserved (other devices)		0	R/O
25	Flash Size (BCM5721, 5751, 5752, 5714, and 5715 only)	Set this bit for a 1-MB device or 0 for a 512-KB device. HARD_RESET, GRC_RESET, and Setting command register bit 0 will reset this bit to pin strap.	pin	R/W
24	Protect Mode (BCM5721, 5751, 5752, 5714, and 5715 only)	Set this bit for flash devices that implement a write protect function. HARD_RESET, GRC_RESET, and Setting command register bit 0 will reset this bit to pin strap.	pin	R/W
23-22	Reserved	-	0	R/O
21-11	SEE_CLK_DIV	This field is a divisor used to create all 1x times for all SEEPROM interface I/O pin timing definitions. A value of 0 means that an SCL transitions at a minimum of each CORE_CLK rising edge. The equation to calculate the clock frequency for SCL is: `CORE_CLK /((SEE_CLK_DIV + 1) * 4)` ***Note:*** SCL is four times slower than 1x time. The default value corresponds to 1.42MHz for BCM5752 and 370 kHz for other devices.	16 for BCM5752 only 44 for other devices	R/W
10-7	SPI_CLK_DIV	This field is a divisor used to create all 1x times for all Flash Interface I/O pin timing definitions. A divisor of 0 means that an SCK transitions at minimum of each CORE_CLK rising edge. The equation to calculate the clock frequency for SCK is: `CORE_CLK /((SPI_CLK_DIV +1) *2)` ***Note:*** SCK is two times slower than 1x time. The default value corresponds to 6.6 MHz.	4	R/W
6-4	Status Bits	This field represents the bit offset in the status command response to interpret as the ready flag.	• 0 if flash_mode = 1 • 7 if buffer_mode = 1 • X otherwise	R/W
3	BitBang_Mode	Enable bit-bang mode to control pins.	0	R/W
2	Pass Mode	Enables pass-through mode to the byte level SPI and SEE state machines.	0	R/W
1	buffer_mode	Enable SSRAM Buffered Interface mode.	pin	R/W
0	flash_mode	Enable Flash Interface mode.	pin	R/W

NVM Config 2 Register (Offset 0x7018)

This register is not applicable to the BCM5700 and BCM5701 MAC Transceivers.

Table 506: NVM Config 2 Register (Offset 0x7018)

Bit	Field	Description	Init	Access
31-24	Reserved	-	0	R/O
23-16	Status Command	This command is used to poll the ready status of the flash device after a command.	• 0x9f if flash_mode = 1 • 0x57 if buffer_mode = 1 • 0x5f if protect_mode = 1	R/W
15-8	Dummy	Value for dummy bytes added to commands.	X	R/W
7-0	Erase Command	Flash device erase command. The ready status is polled after this command.	• 0x20 if flash_mode = 1 • 0x81 if buffer_mode = 1 • 0xd8h if protect_mode = 1	R/W

NVM Config 3 Register (Offset 0x701C)

This register is not applicable to the BCM5700 and BCM5701 MAC Transceivers.

Table 507: NVM Config 3 Register (Offset 0x701C)

Bit	Field	Description	Init	Access
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. Bits[26:25] are address bits A1 and A0 of SEEPROM. User should modify those two bits based on the value of A1 and A0 assigned to this SEEPROM device.	• 0xFF if flash_mode = 1 • 0x68 if buffer_mode = 1 • 0x03 if protect_mode = 1 • 0xA1 otherwise	R/W
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.	0x84 if buffer_mode = 1	R/W
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 SEEPROM (flash_mode=0), this is SEEPROM write command. Bits[10:9] are address bits A1 and A0 of SEEPROM. User should modify those two bits based on the value of A1 and A0 assigned to this SEEPROM device.	• 0x10 if flash_mode = 1 • 0x83 if buffer_mode = 1 • 0xA0 otherwise	R/W
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 be executed.	0x53 if buffer_mode = 1	R/W

SOFTWARE ARBITRATION REGISTER (OFFSET 0X7020)

This register is not applicable to the BCM5700 MAC and BCM5701 MAC Transceivers. This register is used to allow multiple software entities access to NVRAM in a controlled fashion and with a predictable priority scheme.

The input signal ARB_REQ and output signal ARB_GNT will be used to chain multiple chip core together. In the multiple core implementation, the input signal ARB_REQ of the first core in the chain will be connected to ground, and the output signal ARB_GNT will be connected to ARB_REQ of next core. When ARB_GNT is high, the arb request of next core will be masked.

Table 508: Software Arbitration Register (Offset 0x7020)

Bit	Field	Description	Init	Access
31-24	Reserved	-	0	R/O
23	REQ5 (BCM5752 only)	Software request bit 5. A 1 in this bit indicates that the request5 is active	0	R/O
22	ARB_WON5 (BCM5752 only)	Arbitration won bit 5(see Bit 8, ARB_WON0).	0	R/O
21	REQ_CLR5 (BCM5752 only)	Write a 1 to this bit to clear the REQ5 bit.	X	WO
20	REQ_SET5 (BCM5752 only)	Write a 1 to this bit to set the REQ5 bit.	X	WO
19	REQ4 (BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	Software request bit 4. A 1 in this bit indicates that the request4 is active	0	R/O
18	ARB_WON4 (BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	Arbitration won bit 4 (see Bit 8, ARB_WON0).	0	R/O
17	REQ_CLR4 (BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	Write a 1 to this bit to clear the REQ4 bit.	X	WO
16	REQ_SET4 (BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	Write a 1 to this bit to set the REQ4 bit.	X	WO
15	REQ3	Software request bit 3. A 1 in this bit indicates that the request3 is active.	0	R/O
14	REQ2	Software request bit 2. A 1 in this bit indicates that the request2 is active.	0	R/O
13	REQ1	Software request bit 1. A 1 in this bit indicates that the request1 is active.	0	R/O
12	REQ0	Software request bit 0. When Req_set0 bit is set, this bit will be set.	0	R/O
11	ARB_WON3	Arbitration won bit 3 (see Bit 8, ARB_WON0).	0	R/O
10	ARB_WON2	Arbitration won bit 2 (see Bit 8, ARB_WON0).	0	R/O
9	ARB_WON1	Arbitration won bit 1 (see Bit 8, ARB_WON0).	0	R/O

Table 508: Software Arbitration Register (Offset 0x7020) (Cont.)

Bit	Field	Description	Init	Access
8	ARB_WON0	When req0 arbitration is won, this bit will be read as 1. When an operation is complete, then Req_clr0 must be written to clear this bit. At that point, the next high priority Arb bit will be set if requested. At any time, only one of the ARB_WON[5:0] bits will be read as a 1. ARB0 has highest priority, and ARB5 has lowest priority.	0	R/O
7	REQ_CLR3	Write a 1 to this bit to clear the REQ3 bit.		WO
6	REQ_CLR2	Write a 1 to this bit to clear the REQ2 bit.		WO
5	REQ_CLR1	Write a 1 to this bit to clear the REQ1 bit.		WO
4	REQ_CLR0	Write a 1 to this bit to clear the REQ0 bit.		WO
3	REQ_SET3	Write a 1 to this bit to set the REQ3 bit.		WO
2	REQ_SET2	Write a 1 to this bit to set the REQ2 bit.		WO
1	REQ_SET1	Write a 1 to this bit to set the REQ1 bit.		WO
0	REQ_SET0	Set Software Arbitration request bit 0. This bit is set by writing a 1.		WO

Note: By convention, the BCM57XX bootcode has the highest priority for NVRAM access. It uses software request 0 followed by host driver software, which uses software request 1. When the host driver software downloads optional firmware to the BCM57XX (see "Firmware Download" on page 162), one of the required steps is to stop the RX RISC CPU. If the RX RISC CPU is executing bootcode and the ARB_WON0 bit is set when the CPU is stopped, the host driver software must also set the REQ_CLR0 bit to release the NVRAM lock held by the bootcode, otherwise, no other software is able to access NVRAM until the BCM57XX is reset.

NVM Access Register (Offset 0x7024)

This register is applicable for the BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only.

Table 509: NVM Access Register (Offset 0x7024)

Bit	Field	Description	Init	Access
31-2	Reserved	-	0	R/O
1	NVM Access Write Enable	When set, allows the NVRAM write command (see the WR bit in the Table 500 on page 550) to be issued even if the NVRAM Write Enable bit of the Mode Control register (see Table 429 on page 502) is 0.	0	R/W
0	NVM Access Enable	When clear, prevents write access to all other NVM registers, except for the Software Arbitration Register (see Table 508 on page 555).	0	R/W

NVM WRITE1 REGISTER (OFFSET 0X7028)

This register is applicable for the BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only.

Table 510: NVM Write1 Register (Offset 0x7028)

Bit	Field	Description	Init	Access
31-24	Reserved	-	0	R/O
23-16	Status Register Data	Data written to the status register for the SST25VF512.	0	R/W
15-8	Write Disable Command	Flash command to be used when a flash device with a write protect function is used. This command is issued by the flash interface state machine through the SPI interface to write-protect the flash.	0x4	R/W
7-0	Write Enable Command	Flash command to be used when a flash device with a write protect function is used. This command is issued by the flash interface state machine through the SPI interface to write-enable the flash.	0x6	R/W

NVM ARBITRATION WATCHDOG TIMER REGISTER (OFFSET 0X702C)

This register is applicable for the BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only.

Table 511: NVM Arbitration Watchdog Timer Register (Offset 0x702C)

Bit	Field	Description	Init	Access
31-28	TPM Arbitration Lock Timer (BCM5752 only)	This field holds a delay in milliseconds until the TPM will be allowed to access the flash, even if the LAN has not yet set REQ0,REQ1,REQ2,REQ3.	1	R/W
	Reserved	-	0	R/O
27-24	TPM Watchdog Timer Value	This field is the value of the TPM Watchdog Timer in milliseconds.	0	R/W
23-6	Reserved	-	0	R/O
5	TPM Arbitration Lock Timer Enable (BCM5752 only)	Enable Arbitration Lock Timer to allow TPM access to flash after timeout if LAN has not already requested flash access.	1	R/W
	Reserved (other devices)		0	R/O
4	TPM Watchdog Timer Enable	When set, enables a watchdog timer that releases the GNT4/REQ4 bits of the Software Arbitration Register (see Table 508 on page 555) upon expiration of watchdog timer. In case of BCM5752, the GNT5/REQ5 bits are also released along with GNT4/REQ4 bits.	0	R/W
3-0	Reserved	-	0	R/O

ADDRESS LOCKOUT BOUNDARY REGISTER (OFFSET 0X7030)

This register is applicable for the BCM5752 only.

Table 512: Address Lockout Boundary Register (Offset 0x7030)

Bit	Field	Description	Init	Access
31-24	Reserved	-	0	R/O
23-0	Address Lockout Boundary	This register holds the physical address boundary between LAN and TPM space in the flash. When the address-based lockout feature is enabled, the LAN shall not be able to access flash addresses beyond this value.	0	R/W

BIST REGISTERS

The below register definitions are applicable to BCM5721, BCM5751, and BCM5752 only.

Table 513: BIST Registers (Applicable to BCM5721, BCM5751, and BCM5752 only)

Address	Description
0x7400	BIST Control Register
0X7404	BIST Status Register

The below register definitions are applicable to BCM5714 and BCM5715 only.

Table 514: BIST Registers (Applicable to the BCM5714 and BCM5715 Only)

Address	Description
0x7400	BIST Mode Register
0x7404	BIST Status Register
0x7408	BIST Control Register

BIST Control Register (Offset 0x7400)

This register definition is applicable to BCM5721, BCM5751, and BCM5752 only.

Table 515: BIST Control Register (Offset 0x7400, BCM5721, BCM5751, and BCM5752 Only)

Bit	Field	Description	Init	Access
31-6	Reserved	-		
5	BIST_SEL	When set, the internal BIST control is selected. When clear, the external control is selected. This bit is overridden by BIST external.	0	R/W
4	BIH	Memory Burn In.	0	R/W
3	BIST_RM	Trim Bit.	0	R/W
2	BIST_CCM	Trim Bit.	0	R/W
1	BIST_EN	Enable Internal BIST.	0	R/W
0	BIST_RESET	BIST reset.	0	R/W

BIST Status Register (Offset 0x7404)

This register definition is applicable to BCM5721, BCM5751, and BCM5752 only.

Table 516: BIST Status Register (Offset 0x7404, BCM5721, BCM5751, and BCM5752 Only)

Bit	Field	Description	Init	Access
31-8	Reserved	-		
9	FB_BIST_FAIL (BCM5752 only)	Status indicating flash buffer BIST test has failed.	0	R/O
8	FB_BIST_DONE (BCM5752 only)	Status indicating flash buffer BIST test has completed.	0	R/O
7	RB_BIST_FAIL	Status indicating BIST test has failed.	0	R/O
6	RB_BIST_DONE	Status indicating BIST has completed when set (RETRY Buffer).	0	R/O
5	MS_BIST_FAIL	Status indicating BIST test has failed.	0	R/O
4	MS_BIST_DONE	Status indicating BIST has completed when set (MISC_BD).	0	R/O
3	TX_BIST_FAIL	Status indicating BIST test has failed.	0	R/O
2	TX_BIST_DONE	Status indicating BIST has completed when set (TXMBUF).	0	R/O
1	RX_BIST_FAIL	Status indicating BIST test has failed.	0	R/O
0	RX_BIST_DONE	Status indicating BIST has completed when set (RXMBUF).	0	R/O

BIST MODE REGISTER (0X7400H)

This register definition is applicable to BCM5714 and BCM5715 only.

Table 517: BIST Mode Register (0x7400H, BCM5714C and BCM5714S Only)

Bit	Field	Description	Init	Access
31-2	Reserved	-	0	R/O
1	Internal mbist mode sel	-	0	R/W
0	Mbist_reset_l	Resets MBIST logic	0	R/W

BIST STATUS REGISTER (0X7404H)

This register definition is applicable to BCM5714 and BCM5715 only.

Table 518: BIST Status Register (0x7404H, BCM5714 and BCM5715 Only)

Bit	Field	Description	Init	Access
31-12		Reserved	0	R/O
11	Ump_rxfio_mbist_fail	-	0	R/O
10	Ump_rxfio_mbist_done	-	1	R/O
9	Ump_txfio_mbist_fail	-	0	R/O
8	Ump_txfio_mbist_done	-	1	R/O
7	Isram_mbist_fail	-	0	R/O
6	Isram_mbist_done	-	1	R/O
5	Rxmbuf_mbist_fail	-	0	R/O
4	Rxmbuf_mbist_done	-	1	R/O
3	Txmbuf_mbist_fail	-	0	R/O
2	Tmbuf_mbist_done	-	1	R/O
1	Miscbd_mbist_fail	-	0	R/O
0	Miscbd_mbist_done	-	1	R/O

BIST CONTROL REGISTER (0X7408H)

This register definition is applicable to BCM5714 and BCM5715 only.

Table 519: BIST Control Register (0x7408H, BCM5714 and BCM5715 Only)

Bit	Field	Description	Init	Access
29	Ump_rxfio_mbist_dmpen	-	0	R/W
28	Ump_rxfio _mbist_dbgen	-	0	R/W
27	Ump_rxfio _mbist_hold_l	-	1	R/W
26	Ump_rxfio_mbist_en	-	0	R/W
25	Ump_txfio_mbist_dmpen	-	0	R/W
24	Ump_txfio_mbist_dbgen	-	0	R/W
23	Ump_txfio _mbist_hold_l	-	1	R/W
22	Ump_txfio_mbist_en	-	0	R/W
21	isram_mbist_dmp_en	-	0	R/W
20	Isram_mbist_dbg_en	-	0	R/W
19	Isram_mbist_hold_l	-	1	R/W
18	Isram_mbist_en	-	0	R/W
17	Rxmbuf_mbist_dmp_en	-	0	R/W
16	Rxmbuf_mbist_dbg_en	-	0	R/W
15	Rxmbuf_mbist_hold_l	-	1	R/W
14	Rxmbuf_rm	-	0	R/W
13	Rxmbuf_ccm	-	0	R/W
12	Rxmbuf_mbist_en	-	0	R/W
11	Txmbuf_mbist_dmp_en	-	0	R/W
10	Txmbuf_mbist_dbg_en	-	0	R/W
9	Txmbuf_mbist_hold_l	-	1	R/W
8	Txmbuf_rm	-	0	R/W
7	Txmbuf_ccm	-	0	R/W
6	Txmbuf_mbist_en	-	0	R/W
5	Miscbd_mbist_dmp_en	-	0	R/W
4	Miscbd_mbist_dbg_en	-	0	R/W
3	Miscbd_mbist_hold_l	-	1	R/W
2	Miscbd_rm	-	0	R/W
1	Miscbd_ccm	-	0	R/W
0	Miscbd_mbist_en	-	0	R/W

UART REGISTERS

All registers in this section are specific to the BCM5721 and BCM5751 only.

DLAB is bit 7 of the Line Control Register (LCR). It enables reading and writing of the Divisor Latch Registers to set the baud rate of the UART. Below is the summary of the UART register map.

Table 520: UART Register Map Summary

DLAB	Address	Description	Init	Access
0	0x7800	Receive Buffer Register (RBR)	0x00	RO
0	0x7800	Transmit Holding Register (THR)	0x00	WO
1	0x7800	Divisor Latch (Low) (DLL)	0x00	R/W
0	0x7804	Interrupt Enable Register (IER)	0x00	R/W
1	0x7804	Divisor Latch (High) (DLH)	0x00	R/W
X	0x7808	Interrupt Identity Register (IIR)	0x01	RO
X	0x7808	FIFO Control Register (FCR)	0x00	WO
X	0x780c	Line Control Register (LCR)	0x00	R/W
X	0x7810	Modem Control Register (MCR)	0x00	R/W
X	0x7814	Line Status Register (LSR)	0x60	RO
X	0x7818	Modem Status Register (MSR)	0x00	RO
X	0x781c	Scratch Register (SCR)	0x00	R/W

UART RECEIVE BUFFER (DLAB=0, R/O) REGISTER (OFFSET 0X7800)

Table 521: UART Receive Buffer (DLAB=0) Register (Offset 0x7800)

Bit	Field	Description	Init	Access
7-0	RBR	The data in this register is valid only if the Data Ready (DR) bit in the Line Status Register (LSR) is set. In the non-FIFO mode, the data in the RBR must be read before the next data arrives, otherwise, it is overwritten, resulting in an overrun error. In FIFO mode, this register accesses the head of the receive FIFO. If the receive FIFO is full and this register is not read before the next data character arrives, then the data already in the FIFO is preserved, but any incoming data is lost and an overrun error occurs.	0x00	R/O

UART Transmit Holding (DLAB=0, W/O) Register (Offset 0x7800)

Table 522: UART Transmit Holding (DLAB=0) Register (Offset 0x7800)

Bit	Field	Description	Init	Access
7-0	THR	THR contains data to be transmitted on the serial output port (out). Data can be written to the THR any time that the THR Empty (THRE) bit of the LSR is set. If FIFOs are not enabled and THRE is set, writing a single character to the THR clears the THRE. Any additional writes to the THR before the Shasta Register Specification THRE is set again, causes the THR data to be overwritten.	0x00	W/O

UART Divisor Latch (Low) (DLAB=1) Register (Offset 0x7800)

Table 523: UART Divisor Latch (Low) (DLAB=1) Register (Offset 0x7800)

Bit	Field	Description	Init	Access
7-0	DLL	The DLH register in conjunction with the DLL register forms a 16-bit, read/write, Divisor Latch register that contains the baud rate divisor for the UART. It is accessed by first setting the DLAB bit in the Line Control register. The output baud rate is equal to the input clock frequency divided by 16 times the value of the baud rate divisor. `Baud= (clock freq) / (16 * divisor)`	0x00	R/W

UART Interrupt Enable (DLAB=0) Registers (Offset 0x7804)

Table 524: UART Interrupt Enable (DLAB=0) Register (Offset 0x7804)

Bit	Field	Description	Init	Access
7-4	Reserved	-	0	W/O
3	EDSSI	Enable Modem Status Interrupt	0	R/W
2	ELSI	Enable Receiver Line Status Interrupt	0	R/W
1	ETBEI	Enable Transmitter Holding Register Empty Interrupt	0	R/W
0	ERBFI	Enable Received Data Available Interrupt Divisor Latch (High) (DLH) (DLAB=1)	0	R/W

UART Divisor Latch High (DLAB=1) Register (Offset 0x7804)

Table 525: UART Divisor Latch High (DLAB=1) Register (Offset 0x7804)

Bit	Field	Description	Init	Access
7-0	DLH	Divisor Latch (High)	0x00	R/W

UART Interrupt Identity Register (Offset 0x7808, R/O)

Table 526: UART Interrupt Identity Register (Offset 0x7808)

Bit	Field	Description	Init	Access
7-6		• 00: FIFO disable • 11: FIFO enable	0x0	R/O
5-4	Reserved	-	0x0	R/O
3-0	Interrupt ID	• 0000: Modem Status changed • 0001: No interrupt pending • 0010: THR empty • 0100: Received Data available • 0110: Receiver Status • 1100: Character Time out	0x1	R/O

UART FIFO Control Register (Offset 0x7808, W/O)

Table 527: UART FIFO Control Register (Offset 0x7808)

Bit	Field	Description	Init	Access
7-6	RCVR trigger	• 00: 1 byte in FIFO • 01: 4 bytes in FIFO • 10: 8 bytes in FIFO • 11: 14 bytes in FIFO	0	W/O
5-4	Reserved	-	0	W/O
3	-	DMA mode	0	W/O
2	-	XMIT FIFO reset	0	W/O
1	-	RCVR FIFO reset	0	W/O
0	-	FIFO enable	0	W/O

UART Line Control Register (Offset 0x780C)

Table 528: UART Line Control Register (Offset 0x780C)

Bit	Field	Description	Init	Access
7	DLAB	Divisor Latch Access Bit	0	R/W
6	Break	Break Control. 1- sends a break signal by holding the sout line low.	0	R/W
5	Stick Parity	Not used	0	R/W
4	EPS	Parity Select bit	0	R/W
3	PEN	Parity Enable	0	R/W
2	STOP	Number of stop bits. • 0: 1 bit • 1: 2 bits	0	R/W
1:0	CLS	Number of bits per character. • 00: 5 bits • 01: 6 bits • 10: 7 bits • 11: 8 bits	0	R/W

UART Modem Control Register (Offset 0x7810)

Table 529: UART Modem Control Register (Offset 0x7810)

Bit	Field	Description	Init	Access
7-5	Reserved	-	0x00	R/W
4	Loopback	1= sout is held high, while serial data output is looped back to the sin line internally.	0	R/W
3	Out2	This bit is inverted to generate out2#.	0	R/W
2	Out1	This bit is inverted to generate out1#.	0	R/W
1	RTS	This bit is inverted to generate RTS#.	0	R/W
0	DTR	This bit is inverted to generate DTR#.	0	R/W

UART Line Status Register (Offset 0x7814)

Table 530: UART Line Status Register (Offset 0x7814)

Bit	Field	Description	Init	Access
7	FERR	RX FIFO Error	0	R/O
6	TEMT	Transmitter Empty	1	R/O
5	THRE	Transmitter Holding Register Empty	1	R/O
4	BI	Break Interrupt	0	R/O
3	FE	Framing Error	0	R/O
2	PE	Parity Error	0	R/O
1	OE	Overrun Error	0	R/O
0	DR	Data Ready	0	R/O

UART Modem Status Register (Offset 0x7818)

Table 531: UART Modem Status Register (Offset 0x7818)

Bit	Field	Description	Init	Access
7	DCD	Compliment of DCD#	0	R/O
6	RI	Compliment of RI#	0	R/O
5	DSR	Compliment of DSR#	0	R/O
4	CTS	Compliment of CTS#	0	R/O
3	DDCD	Record DCD# change since last MSR read	0	R/O
2	TERI	Record RI# change since last MSR read	0	R/O
1	DDSR	Record DSR# change since last MSR read	0	R/O
	DCTS	Record CTS# change since last MSR read	0	R/O

UART Scratch Register (Offset 0x781C)

Table 532: UART Scratch Register (Offset 0x781C)

Bit	Field	Description	Init	Access
7:0	SCR	Scratch Register	0x00	R/W

UMP REGISTERS

All registers in this section are applicable to BCM5714C, BCM5714S, BCM5715C, and BCM5715S only.

Table 533: UMP Registers (Applicable to BCM5714 and BCM5715 Only)

Address	Description
0x7800	UMP Attention Enable register
0x7804	UMP Attention Status register
0x7808	UMP Debug1 register
0x780C	UMP Debug2 register
0x7810	UMP Command register
0x7814	UMP Status register
0x7818	UMP FRM Read Status register
0x781C	UMP FRM Read Data register
0x7820	UMP FRM Write Control register
0x7824	UMP FRM Write Data register
0x7828	UMP Frame Pre-fetch register
0x782C	UMP FIFO Remain register

UMP ATTENTION ENABLE REGISTER (OFFSET 0X7800)

Table 534: UMP Attention Enable Register (Offset 0x7800, BCM5714 and BCM5715 Only)

Bit	Field	Description	Init	Access
31:6	Reserved	Returns zeros	0	R/W
5	Txrdy_en	Enable TX Ready event	1	R/W
4	Rxrdy_en	Enable RX Ready event	1	R/W
3	Fiotx_Perr_en	Enable parity error from FIO TX Buffer	0	R/W
2	Fiorx_Perr_en	Enable parity error from FIO RX Buffer	0	R/W
1	Mactx_Perr_en	Enable parity error from MAC TX Buffer	0	R/W
0	Macrx_Perr_en	Enable parity error from MAC RX Buffer	0	R/W

UMP ATTENTION STATUS REGISTER (OFFSET 0X7804)

Table 535: UMP Attention Status Register (Offset 0x7804, BCM5714 and BCM5715 Only)

Bit	Field	Description	Init	Access
31:4	Reserved	Returns zeros	0	R/W
3	Fiotx_Perr_detected	Parity error from FIO TX Buffer detected	0	W2C
2	Fiorx_Perr_detected	Parity error from FIO RX Buffer detected	0	W2C
1	Mactx_Perr_detected	Parity error from MAC TX Buffer detected	0	W2C
0	Macrx_Perr_detected	Parity error from MAC RX Buffer detected	0	W2C

UMP DEBUG1 REGISTER (OFFSET 0X7808)

Table 536: UMP Debug1 Register (Offset 0x7808, BCM5714 and BCM5715 Only)

Bit	Field	Description	Init	Access
31:9	Reserved	Returns zeros	0	R/W
8	UMP MAC Loop back en	Internal loop back mode	0	R/W
7:4	Debug2_mux_sel	Mux select for UMP Debug vector 2	0	R/W
3:0	Debug1_mux_sel	Mux select for UMP Debug vector 1	0	R/W

UMP COMMAND REGISTER (OFFSET 0X7810)

Table 537: UMP Command Register (Offset 0x7810, BCM5714 and BCM5715 Only)

Bit	Field	Description	Init	Access
31:16	Reserved	Returns zeros.	0	R/W
15	Cfg_no_alignment	If set, core does not do the 2-byte alignment required by Teton.	0	R/W
14	Rst_tx_state	When set, this allows the TX state machine to be reset even after a byte count was programmed in. This makes it possible to throw away a frame if there is not enough TX buffer space and start all over again.	0	R/W
13	Cfg_tx_drive	When set, the core will generate a txdrive pulse one clock before TXEN and one clock after TXEN.	0	R/W
12:10	Reserved	Returns zeros.	0	R/W
9	Sw_pause	When this bit is1, it forces the pause needed input to the MAC to 1. This may be used by FW to force pause packets to be generated during times when it knows that it cannot process received packets properly.	0	R/W
8	Tx_drive	When this bit is 1, it enables the TXD, TXEN, and TXER pins of the UMP interface to drive. When this bit is 0, the outputs float. This bit must be set to 1 for transmit data to be driven. The value of this bit has no effect on the internal FIFO or MAC operation.	0	R/W
7	Tx_drop	When this bit is set, it indicates that UMP MAC will drop all new TX packets. Any current packet being transmitted will be completed.	1	R/W
6	TX_mac_disable	When this bit is set, it disables the transmit function of the UMP MAC. Setting this bit to 1 will reset the UMP port transmitter to a known state. This bit must be set to 0 to allow transmission.	1	R/W
5	Rx_mac_disable	When this bit is set, it disables the receive function of the UMP MAC. Setting this bit to 1 will reset the UMP port receiver to a known state. This bit must be set to 0 to allow reception of new frames.	0	R/W
4	Hdflowsel	This bit enables the generation of a 3000 byte TX frame, with bad CRC, as long as the RX FIFO is full. When this bit is low, then the FLOWMODE bit controls how the RX FIFO is monitored.	0	R/W

Table 537: UMP Command Register (Offset 0x7810, BCM5714 and BCM5715 Only) (Cont.)

Bit	Field	Description	Init	Access
3	Flowmode	This bit provides a fast pause packet generation mode where the current TX packet is truncated with bad CRC to allow the pause packet to be generated. If this bit is 0, then any current TX packet will complete normally before any pause packet is generated.	0	R/W
2	Fc_en	When this bit is 1, it indicates that flow control is enabled. If this bit is 0, no attempt will be made to prevent packet drop. When this bit is 1 and HDFLOWSEL is 0, pause packets will be generated to attempt to prevent drop of RX packets.	0	R/W
1	Tx_fifo_enabled	This bit controls the current enable status of the TX side of the UMP FIFO block. If this bit is 1, it indicates that the TX FIFO is enabled. When this bit is 0, the FIFO will be reset immediately. If the transmit MAC is currently transmitting a packet and does not get all the data needed for the packet, it will underrun and transmit JAM for the packet and then stop.	0	R/W
0	Rx_fifo_enabled	This bit controls the current enable status of receive UMP FIFO block. If this bit is 1, it indicates that the RX FIFO is enabled. When this bit is 0, the reset FIFO will be reset immediately. Any partial or complete packets in the FIFO will be lost. This bit has no effect on the reception of frames by the RX MAC. Use RX_DISABLE for that purpose. UMP Reset Conditions: • CPU_CLK Logic in UMP is reset if TX_FIFO_ENABLE, RX_FIFO_ENABLE are both zero and HDFLOWSEL is set. • CORE_CLK Logic in UMP is reset if TX_FIFO_ENABLE, RX_FIFO_ENABLE are both zero and HDFLOWSEL is set. • RX_CLK logic in UMP is reset if RX_DISABLE AND FC_EN are both set. • TX_CLK logic in UMP is reset if TX_DISABLE AND HDFLOWSEL are both set.	0	R/W

UMP Status Register (Offset 0x7814)

Table 538: UMP Status Register (Offset 0x7814, BCM5714 and BCM5715 Only)

Bit	Field	Description	Init	Access
31:24	Xon_trip	When the RX FIFO has at least XON_TRIP*16 bytes available, the pause needed input to the UMP MAC will be de-asserted. This value is the high-watermark for flow control generated by the UMP MAC. A value is the largest valid value that may be programmed and will de-assert the pause needed signal only when the RX FIFO is empty.	0x96	R/W
23:16	Xoff_trip	When the RX FIFO has less than XOFF_TRIP*16 bytes remaining, the pause needed input to the UMP MAC will be asserted. This value is the low-watermark for flow control generated by the UMP MAC. If this value is 0, then the pause needed input to the UMP MAC will be controlled solely by the SW_PAUSE bit.	0x48	R/W
15:6	Reserved		0	R/W
5	Src_addr_chg	SRC_ADDR_CHG When this bit is set, it indicates that the link source address has been changed since last read operation. The bit remains set until a 1 is written to this bit position.	0	W2C
4	Reserved		0	R/W
3	Rx_frm_drop	When this bit is set, it indicates that RX packets have been dropped due to RX FIFO full. This bit is cleared when a 1 is written to this bit position.	0	W2C
2	Reserved		0	R/W
1	Fdx	This bit sets the duplex mode to be used by the UMP MAC. When this bit is 0, half-duplex Ethernet protocol will be followed. When this bit is 1, then full-duplex operation will be enabled.	0	R/W
0	Tx_idle	This bit reflects the current transmit status of the UMP port. It will read as 1 if there is transmit data pending in the TX FIFO section or if the TX MAC is currently transmitting a packet. When this bit is zero, then the transmitter is completely idle. This bit is intended to be sampled after the TX_FIFO_ENABLED bit is cleared. After this bit reads as 0, the TX_DRIVE output can be turned off.	0	R/W

UMP Frame Read Status Register (Offset 0x7818)

Table 539: UMP Frame Read Status Register (Offset 0x7818, BCM5714 and BCM5715 Only)

Bit	***Field***	***Description***	***Init***	***Access***
31:29	Rx_fifo_state	0: Idle 1: Ready 2: Busy 3: EXTRA_RD Extra Read 4: LATCH_IP_HDR	0	R
28:14	Reserved		0	R
13:3	Frm_rd_st_bcnt	Current RX packet length in byte count.	0	R
2	Frm_rd_sta_fifo_empty	When this bit is set, it indicates that RX FIFO is empty and firmware or software should not read rd_dat any further until this bit clears.	1	R
1	Frm_rd_sta_frm_in_pro	When this bit is set, it indicates that current packet reading is in progress. Cleared when all packet data has been read for a particular packet.	0	R
0	Frm_rd_sta_new_frm	When this bit is set, it indicates that a new packet is ready in RX FIFO. Always 0 when FRM_IN_PRO bit is set. This bit maps directly to the value of the FTQ2_VALID event bit.	0	R

UMP Frame Read Data Register (Offset 0x781C)

Table 540: UMP Frame Read Data Register (Offset 0x781C, BCM5714 and BCM5715 Only)

Bit	***Field***	***Description***	***Init***	***Access***
31:0	Data	Next 32-bit data word of a packet. For a new packet, the first two reads of this register contain dummy data. On the third read, the right two bytes contain the first two data bytes of the packet. This means that the 3rd byte of the packet is in bits 31:24 of the 4th data word read. The two dummy reads are required to discard control data for the packet. The two extra bytes in the 3rd read makes the L3 and above frame headers 32-bit aligned with the frm_rd_data register.	0	R/O

UMP Frame Write Control Register (Offset 0x7820)

Table 541: UMP Frame Write Control Register (Offset 0x7820, BCM5714 and BCM5715 Only)

Bit	Field	Description	Init	Access
31:30	Tx_fifo_state	0: Idle 1: Wait 2: Busy 3: EXTRA_WR	0	R
29:14	Reserved		0	R
13:3	Frm_wr_ctrl_bcnt	This register must be written with the packet length before the ump_frm_wr_dat register at offset 0x7824 is written with any data. For this value to be accepted, the SOF bit must be set on the same register write.	0	R
2	Frm_wr_ctrl_bcnt_rdy	When this bit is set, it indicates that the BCNT field can accept another write as a request for space for another transmit frame. This bit is logically ORed with the FIFO_RDY bit to create the value of the FTQ1_VALID event bit. This bit must be 1 when the firmware writes a 1 to the NEW_FRM bit. This bit will clear back to 0 each time a new BCNT value is written and will only set back to 1 once the NEW_FRM bit has cleared at the end of the complete packet write.	1	R
1	Frm_wr_ctrl_fifo_rdy	This bit is set when the TX_FIFO can accept all the data requested by a write to the BCNT value. This value will read as 1 when there is sufficient space in the TX FIFO to accept all the bytes described by the latest BCNT value. This bit will clear to 0 once all the data requested in BCNT has been written to the ump_frm_wr_data register. This bit is logically ORed with the BCNT_RDY bit to create the value of the FTQ1_VALID event bit.	0	R
0	Frm_wr_ctrl_new_frm	This bit must be set for the BCNT field to be accepted as the length of a new packet. This bit will clear to zero when the complete packet has been written to the ump_frm_wr_data register at offset 0x7824.	0	R

UMP Frame Write Data Register (Offset 0x7824)

Table 542: UMP Frame Write Data Register (Offset 0x7824, BCM5714 and BCM5715 Only)

Bit	Field	Description	Init	Access
31:0	Data	Next 32-bit data word of a transmit packet. Only the right two bytes of the first data word of each packet are valid. This means that the 3rd byte of the packet is in bits 31:24 of the 2nd data word write. This offset of 2 bytes makes the L3 and above frame headers 32 -bit aligned with the rm_wr_data register. This register must only be written when FIFO_RDY bit is 1 in the ump_frm_wr_ctrl register at offset 0x7820.	0	R/W

UMP Frame Pre-Fetch Register (Offset 0x7828)

Table 543: UMP Frame Pre-fetch Register (Offset 0x7828, BCM5714 and BCM5715 Only)

Bit	Field	Description	Init	Access
31:0	Data	This register provides the ethernet type field of the packet when a new packet is ready to be read from the rd_data register. Reading this register has no effect on the FIFO status. The value of this register is invalid except before the first data is read from the rd_data register at offset 0x781C.	0	R/O

UMP FIFO Remain Register (Offset 0x782C)

Table 544: UMP FIFO Remain Register (Offset 0x782C, BCM5714 and BCM5715 Only)

Bit	Field	Description	Init	Access
31:25	Reserved	-	0	R
24:16	RxFifo_byte_cnt	Bytes Remaining in RX FIFO		R
15:9	Reserved	-	0	R
8:0	TxFifo_byte_cnt	Bytes Remaining in TX FIFO		R

PCIE REGISTERS

These registers are applicable to the BCM5721, BCM5751, and BCM5752 only.

Table 545: PCIe Registers

Address	***Description***
TLP Diagnostic Registers (these are Transaction Layer Protocol Hardware Debug Registers)	
0x7C00	TLP Control register
0x7C04	TLP Workaround Register
0x7C08	Reserved
0x7C0C	Reserved
0x7C10	Write DMA Request Upper Address Diagnostic register
0x7C14	Write DMA Request Lower Address Diagnostic register
0x7C18	Write DMA Length/Byte Enable and Request Diagnostic register
0x7C1C	Read DMA Request Upper Address Diagnostic register
0x7C20	Read DMA Request Lower Address Diagnostic register
0x7C24	Read DMA Length and Request Diagnostic register
0x7C28	MSI DMA Request Upper Address Diagnostic register
0x7C2C	MSI DMA Request Lower Address Diagnostic register
0x7C30	MSI DMA Length and Request Diagnostic register
0x7C34	Slave Request Length and Type Diagnostic register
0x7C38	Flow Control Inputs Diagnostic register
0x7C3C	XMT State Machines and Gated Requests Diagnostic register
0x7C40	Address ACK Xfer Count and ARB Length Diagnostic register
0x7C44	DMA Completion Header Diagnostic register 0
0x7C48	DMA Completion Header Diagnostic register 1
0x7C4C	DMA Completion Header Diagnostic register 2
0x7C50	DMA Completion Misc. Diagnostic register
0x7C54	DMA Completion Misc. Diagnostic register
0x7C58	DMA Completion Misc. Diagnostic register
0x7C5C	Split Controller Requested Length and Address ACK Remaining Diagnostic register
0x7C60	Split Controller Misc 0 Register Diagnostic register
0x7C64	Split Controller Misc 1 Register Diagnostic register
0x7C68	TLP Register BusNo, DevNo, FuncNo Register
0x7C6C	TLP Debug Register
Data Link Layer Internal Registers	
0x7D00	Data Link Control register
0x7D04	Data Link Status register
0x7D08	Data Link Attention register

Table 545: PCIe Registers (Cont.)

Address	***Description***
0x7D0C	Data Link Attention Mask register
0x7D10	Next Transmit Sequence Number Debug register
0x7D14	Acked Transmit Sequence Number Debug register
0x7D18	Purged Transmit Sequence Number Debug register
0x7D1C	Receive Sequence Number Debug register
0x7D20	Data Link Replay register
0x7D24	Data Link ACK Timeout register
0x7D28	Power Management Threshold register
0x7D2C	Retry Buffer Write Pointer Debug register
0x7D30	Retry Buffer Read Pointer Debug register
0x7D34	Retry Buffer Purged Pointer Debug register
0x7D38	Retry Buffer Read/Write Debug Port
0x7D3C	Error Count Threshold register
0x7D40	TLP Error Counter register
0x7D44	DLLP Error Counter
0x7D48	NAK Received Counter
0x7D4C	Data Link Test register
0x7D50	Packet BIST register
0x7D54-0x7dff	Reserved
PHY Internal Registers	
0x7E00	PHY Mode register
0x7E04	PHY/Link Status register
0x7E08	PHY/Link LTSSM Control register
0x7E0C	PHY/Link Training Link Number
0x7E10	PHY/Link Training Lane Number
0x7E14	PHY/Link Training N_FTS
0x7E18	PHY Attention Register
0x7E1C	PHY Attention Mask register
0x7E20	PHY Receive Error Counter
0x7E24	PHY Receive Framing Error Counter
0x7E28	PHY Receive Error Threshold register
0x7E2C	PHY Test Control register
0x7E30	PHY/SerDes Control Override
0x7E34	PHY Timing Parameter Override
0x7E38	PHY Hardware Diagnostic1 n TX/RX SM States
0x7E3C	PHY Hardware Diagnostic2 n LTSSM States

TLP Control Register (Offset 0x7C00)

This register is applicable to the BCM5721, BCM5751, and BCM5752 only.

Table 546: TLP Control Register (Offset 0x7C00, BCM5721, BCM5751, and BCM5752 Only)

Bit	Field	Description	Init	Access
31	Enable Excessive current Fix (B1 or higher version of BCM5721 and BCM5751 devices)	Enable the DOS excessive current fix: • 0 = Disable fix • 1 = Enable fix ***Note:*** Refer to E6_5751B0_10362 in the *5751-ES4xx-R Errata* for more details.	1	R/W
30	Reserved	-	0	R/O
29	Enable Interrupt Mode Fix (BCM5721 and BCM5751 only)	Enable the Interrupt Mode Fix: • 0 = Disable fix • 1 = Enable fix ***Note:*** Refer to E3_5751B0_9804 in the *5751-ES-4xx-R Errata* for more details.	0	R/W
28	Reserved	-	0	R/O
27	Enable Unexpected Completion Error Fix	Enable the Unexpected Completion Error Fix: • 0 = Disable fix • 1 = Enable fix The hardware fix is to not send the Unexpected Completion Error message when the chipset replays a completion packet because of BIOS not programming the chipset's replay timer correctly.	0	R/W
26	Enable Type1 Vendor Defined Message Fix	Enable the Type 1 Vendor Defined Message Fix: • 0 = Disable fix • 1 = Enable fix The fix for this is to discard any Type1 message with a data payload of two or more DWs to prevent the data FIFO from getting out of sync.	0	R/W
25	Data FIFO Protect	When set, this bit enables Data FIFO protection.	0	R/W
24	Enable Address Check	When set, this bit enables Address and Type field checking in the Transaction Layer Packet (TLP).	1	R/W
23	Enable TC0 Check	When set, this bit enabled TC0 Traffic Class checking in the TLP.	1	R/W
22	CRC Swap	When set, this bit enables swapping of the digest field when ECRC is enabled.	0	R/W
21	Disable CA Error	When clear, this bit enables the DMA completion logic to check for a completion packet with a Completer Abort Completion Status value.	0	R/W
20	Disable UR Error	When clear, this bit enables the DMA completion logic to check for a completion packet with an Unsupported Request value.	0	R/W
19	Disable RSV Error	When clear, this bit enables the DMA completion logic to check for a completion packet with a Reserved value.	0	R/W
18	Enable MPS Check	When set, this bit enables the DMA completion logic to check for a TLP that violates the Maximum Payload Size requirement.	1	R/W

Table 546: TLP Control Register (Offset 0x7C00, BCM5721, BCM5751, and BCM5752 Only)

Bit	Field	Description	Init	Access
17	Disable EP Error	When clear, this bit enables the Transaction Layer to check for Data Poisoning.	0	R/W
16	Enable Bytecount Check	When set, this bit enables the Transaction Layer's target to check for byte count errors on incoming target accesses.	1	R/W
15-14	Reserved	-	0	R/W
13-11	DMA Read Traffic Class	DMA Read Traffic Class.	0x0	R/W
10-8	DMA Write Traffic Class	DMA Write Traffic Class.	0x0	R/W
7-6	Reserved	-	0x0	R/W
5-0	Completion Timeout	Programmable completion timeout in milliseconds.	0x2f	R/W

TLP WORKAROUND REGISTER (OFFSET 0X7C04)

This register is applicable to BCM5721, BCM5751, and BCM5752 only.

Table 547: TLP Workaround Register (Offset 0x7C04, BCM5752 Only)

Bit	Field	Description	Init	Access
31-9	Reserved	–	0	R/O
8	Enable Training Error Fix	• 0 = Disabled • 1 = Enabled Enable this fix to mask of the training error to avoid any false triggering of this fatal error. The false triggering may be possible as the training error bit is not well defined.	0	R/W
7	Enable GPHY DLL Issue Fix	• 0 = Disabled • 1 = Enabled Enable this fix to automatically detect the GPHY DLL power-down and switch to Slow Core Clk mode to enable the access of EMAC registers when the GPHY DLL is powered down.	1	R/W
6-4	Reserved	–	000b	R/W
3	Enable UR Status Bit Fix	• 1 = UR status bit in the device status register is set if a memory read or write occurs to an unmatched base address. • 0 = UR status bit is not affected by writes to an unmatched base address.	0	R/W
2	Enable Vendor Defined Message Fix	• 1 = UR status bit in the device status register is not set for routing codes 000b, 010b, 011b, and 100b. • 0 = UR status bit in the device status register is set for routing codes 000b, 010b, 011b, and 100b.	0	R/W
1	power_state_write_mem_enable	• 1 = Software can place the device in D3HOT state via a memory write or a configuration write cycle. • 0 = Software can place the device in D3HOT only via a configuration write.	1	R/W
0	Enable Max Payload Size Fix	• 1 = Enable fix for 512-byte MPS • 0 = Disable fix for 512-byte MPS Enable this fix for operating with MPS larger than 128 bytes. Refer to E2_5751B0_9709 in the *5751-ES4xx-R* errata for more details.	1	R/W

WRITE DMA REQUEST UPPER ADDRESS DIAGNOSTIC REGISTER (OFFSET 0X7C10)

This register is applicable to the BCM5721, BCM5751, and BCM5752 only.

Table 548: Write DMA Request Upper Address Diagnostic Register (Offset 0x7C10)

Bit	Field	Description	Init	Access
31-0	Reg_dw_upr_addr	Write DMA Request Upper Address (63:32)	0	R/O

WRITE DMA REQUEST LOWER ADDRESS DIAGNOSTIC REGISTER (OFFSET 0X7C14)

This register is applicable to the BCM5721, BCM5751, and BCM5752 only.

Table 549: Write DMA Request Lower Address Diagnostic Register (Offset 0x7C14)

Bit	Field	Description	Init	Access
31-0	Reg_dw_lwr_addr	Write DMA Request Lower Address (31:0)	0	R/O

WRITE DMA LENGTH/BYTE ENABLE AND REQUEST DIAGNOSTIC REGISTER (OFFSET 0X7C18)

This register is applicable to the BCM5721, BCM5751, and BCM5752 only.

Table 550: Write DMA Length/Byte Enable and Request Diagnostic Register (Offset 0x7C18)

Bit	Field	Description	Init	Access
31-16	Reg_dw_lenbe_req	Write DMA Request Length(15:0)	0	R/O
15-8	-	Write DMA Request Byte Enables(7:0)	0	R/O
7-1	-	Reserved	0	R/O
0	-	Write DMA Raw Request	0	R/O

READ DMA REQUEST UPPER ADDRESS DIAGNOSTIC REGISTER (OFFSET 0X7C1C)

This register is applicable to the BCM5721, BCM5751, and BCM5752 only.

Table 551: Read DMA Request Upper Address Diagnostic Register (Offset 0x7C1C)

Bit	Field	Description	Init	Access
31-0	Reg_dr_upr_addr	Read DMA Request Upper Address (63:32)	0	R/O

READ DMA REQUEST LOWER ADDRESS DIAGNOSTIC REGISTER (OFFSET 0X7C20)

This register is applicable to BCM5721, BCM5751, and BCM5752 only.

Table 552: Read DMA Request Lower Address Diagnostic Register (Offset 0x7C20)

Bit	Field	Description	Init	Access
31-0	Reg_dr_lwr_addr	Read DMA Request Lower Address (31:0)	0	R/O

READ DMA LENGTH AND REQUEST DIAGNOSTIC REGISTER (OFFSET 0X7C24)

This register is applicable to BCM5721, BCM5751, and BCM5752 only.

Table 553: Read DMA Length and Request Diagnostic Register (Offset 0x7C24)

Bit	Field	Description	Init	Access
31-16	Reg_dr_len_req	Read DMA Request Length(15:0)	0	R/O
15-1	Reserved	-	0	R/O
0	-	Read DMA Raw Request	0	R/O

MSI DMA REQUEST UPPER ADDRESS DIAGNOSTIC REGISTER (OFFSET 0X7C28)

This register is applicable to BCM5721, BCM5751, and BCM5752 only.

Table 554: MSI DMA Request Upper Address Diagnostic Register (Offset 0x7C28)

Bit	Field	Description	Init	Access
31-0	Reg_msi_upr_addr	MSI DMA Request Upper Address (63:32)	0	R/O

MSI DMA REQUEST LOWER ADDRESS DIAGNOSTIC REGISTER (OFFSET 0X7C2C)

This register is applicable to BCM5721, BCM5751, and BCM5752 only.

Table 555: MSI DMA Request Lower Address Diagnostic Register (Offset 0x7C2C)

Bit	Field	Description	Init	Access
31-0	Reg_msi_lwr_addr	MSI DMA Request Lower Address (31:0)	0	R/O

MSI DMA Length and Request Diagnostic Register (Offset 0x7C30)

This register is applicable to BCM5721, BCM5751, and BCM5752 only.

Table 556: MSI DMA Length and Request Diagnostic Register (Offset 0x7C30)

Bit	Field	Description	Init	Access
31-16	Reg_msi_len_req	MSI DMA Request Length(15:0)	0	R/O
15-1	Reserved	-	0	R/O
0	-	MSI DMA Raw Request	0	R/O

Slave Request Length and Type Diagnostic Register (Offset 0x7C34)

This register is applicable to BCM5721, BCM5751, and BCM5752 only.

Table 557: Slave Request Length and Type Diagnostic Register (Offset 0x7C34)

Bit	Field	Description	Init	Access
31-26	Reg_slv_len_req	Reserved	0	R/O
25-16	-	Slave Request Length (9:0)	0	R/O
15-2	Reserved	-	0	R/O
1	-	Slave Request Type: 0: Msg 1: Target	0	R/O
0	-	Slave Raw Request	0	R/O

Flow Control Inputs Diagnostic Register (Offset 0x7C38)

This register is applicable to BCM5721, BCM5751, and BCM5752 only.

Table 558: Flow Control Inputs Diagnostic Register (Offset 0x7C38)

Bit	Field	Description	Init	Access
31-27	Reg_fc_input	Reserved	0	R/O
26	-	Flow Control Non-Posted Header Available	0	R/O
25	-	Flow Control Posted Header Available	0	R/O
24	-	Flow Control Completion Header Available	0	R/O
23-12	-	Flow Control Posted Data Available	0	R/O
11-0	-	Flow Control Completion Data Available	0	R/O

XMT State Machines and Gated Requests Diagnostic Register (Offset 0x7C3C)

This register is applicable to BCM5721, BCM5751, and BCM5752 only.

Table 559: XMT State Machines and Gated Requests Diagnostic Register (Offset 0x7C3C)

Bit	Field	Description	Init	Access
31	Reg_sm_r0_r3	Reserved	0	R/O
30-28	-	TLP Transmitter Data State Machine	0	R/O
27-24	-	TLP Transmitter Arbitration State Machine	0	R/O
23-4	-	Reserved	0	R/O
3	-	Slave DMA Gated Request.	0	R/O
2	-	MSI DMA Gated Request	0	R/O
1	-	Read DMA Gated Request	0	R/O
0	-	Write DMA Gated Request	0	R/O

Address ACK Xfer Count and ARB Length Diagnostic Register (Offset 0x7C40)

This register is applicable to BCM5721, BCM5751, and BCM5752 only.

Table 560: Address ACK Xfer Count and ARB Length Diagnostic Register (Offset 0x7C40)

Bit	Field	Description	Init	Access
31-16	-	Address ACK Transfer Count	0	R/O
15-0	-	Arbitration Length	0	R/O

DMA Completion Header Diagnostic Register 0 (Offset 0x7C44)

This register is applicable to BCM5721, BCM5751, and BCM5752 only.

Table 561: DMA Completion Header Diagnostic Register 0 (Offset 0x7C44)

Bit	Field	Description	Init	Access
31-0	Reg_hdr0	DMA Completion Header 0	0	R/O

DMA Completion Header Diagnostic Register 1 (Offset 0x7C48)

This register is applicable to BCM5721, BCM5751, and BCM5752 only.

Table 562: DMA Completion Header Diagnostic Register 1 (Offset 0x7C48)

Bit	Field	Description	Init	Access
31-0	Reg_hdr0	DMA Completion Header 1	0	R/O

DMA Completion Header Diagnostic Register 2 (Offset 0x7C4C)

This register is applicable to BCM5721, BCM5751, and BCM5752 only.

Table 563: DMA Completion Header Diagnostic Register 2 (Offset 0x7C4C)

Bit	Field	Description	Init	Access
31-0	Reg_hdr0	DMA Completion Header 2	0	R/O

DMA Completion Misc Diagnostic Register (Offset 0x7C50)

This register is applicable to BCM5721, BCM5751, and BCM5752 only.

Table 564: DMA Completion Misc. Diagnostic Register (Offset 0x7C50)

Bit	Field	Description	Init	Access
31-28	Reg_dma_cmpt_misc0	Completion Data Poisoning Error Counter	0	R/O
27-24	-	Completion Unexpected Error Counter	0	R/O
23-20	-	Completion CA Error Counter	0	R/O
19-16	-	Completion UR Error Counter	0	R/O
15-12	-	Completion with CRC Error Counter	0	R/O
11-8	-	Completion with Bad TLP Error Counter	0	R/O
7-4	-	Not Used	0	R/O
3-0	-	Completion State Machine	0	R/O

DMA Completion Misc Diagnostic Register (Offset 0x7C54)

This register is applicable to BCM5721, BCM5751, and BCM5752 only.

Table 565: DMA Completion Misc. Diagnostic Register (Offset 0x7C54)

Bit	Field	Description	Init	Access
31-28	Reg_dma_cmpt_misc0	Completion Malform Error Counter	0	R/O
27-24	-	Frame Stay Too Long Error Counter	0	R/O
23-20	-	Frame Drop Too Early Error Counter	0	R/O
19-16	-	Type_Status_Mismatch Error Counter	0	R/O
15-12	-	Status_Malform Error Counter	0	R/O
11-8	-	Completion RCB Error Counter	0	R/O
7-4	-	Completion ByteCount Error Counter	0	R/O
3-0	-	Completion MPS Error Counter	0	R/O

DMA Completion Misc Diagnostic Register (Offset 0x7C58)

This register is applicable to BCM5721, BCM5751, and BCM5752 only.

Table 566: DMA Completion Misc. Diagnostic Register (Offset 0x7C58)

Bit	Field	Description	Init	Access
31-8	Reg_dma_cmpt_misc0	Not Used	0	R/O
7-4	-	Completion Too Much Data Error Counter	0	R/O
3-0	-	Frame Dead-Time Error Counter	0	R/O

Split Controller Requested Length and Address ACK Remaining Diagnostic Register (Offset 0x7C5C)

This register is applicable to BCM5721, BCM5751, and BCM5752 only.

Table 567: Split Controller Requested Length and Address ACK Remaining Diag. Reg. (Offset 0x7C5C)

Bit	Field	Description	Init	Access
31-27	Reg_len_ack_remain	Reserved	0	R/O
26-16	-	Requested Length	0	R/O
15-11	-	Reserved	0	R/O
10-0	-	Address ACK remaining	0	R/O

Split Controller Misc 0 Register Diagnostic Register (Offset 0x7C60)

This register is applicable to BCM5721, BCM5751, and BCM5752 only.

Table 568: Split Controller Misc 0 Register Diagnostic Register (Offset 0x7C60)

Bit	Field	Description	Init	Access
31-18	Reg_splitctl_misc0	Reserved	0	R/O
17	-	Split Pending	0	R/O
16	-	Split Pending Block Request	0	R/O
15-8	-	Initiator Tag	0	R/O
7	-	Reserved	0	R/O
6	-	Split Controller Timeout Status	0	R/O
5-0	-	Split Controller Timeout Counter	0	R/O

SPLIT CONTROLLER MISC 1 REGISTER DIAGNOSTIC REGISTER (OFFSET 0X7C64)

This register is applicable to BCM5721, BCM5751, and BCM5752 only.

Table 569: Split Controller Misc 1 Register Diagnostic Register (Offset 0x7C64)

Bit	Field	Description	Init	Access
31-29	Reg_splitctl_misc1	Reserved	0	R/O
28-26	-	Registered Traffic Class	0	R/O
25-24	-	Registered Attribute	0	R/O
28-8	-	Registered Requester ID	0	R/O
7-0	-	Registered Initiator Tag	0	R/O

TLP BUS, DEV, AND FUNC NUMBER REGISTER (OFFSET 0X7C68, BCM5721, BCM5751, AND BCM5752 ONLY)

Table 570: TLP Status Register (Offset 0x7C60)

Bit	Field	Description	Init	Access
31-17	Reserved	Reserved	0	R/O
16	Config Write Indicator	First config write has been received	0	R/O
15-8	Bus number	PCI Bus Number	0	R/O
7-3	Device number	PCI Device Number	0	R/O
2-0	Function number	PCI Function number	0	R/O

TLP DEBUG REGISTER (OFFSET 0X7C6C, BCM5721, AND BCM5751 ONLY)

Table 571: TLP Status Register (Offset 0x7C60)

Bit	Field	Description	Init	Access
31	A4 Device Indication Bit	Indicates whether the device is A4 chip	1 for A4, 0 for other versions	R/O
30	B1 Device Indication Bit	Indicates whether the device is B1 chip	1 for B1, 0 for other versions.	R/O
29-0	Reserved		0	R/O

DATA LINK CONTROL REGISTER (OFFSET 0X7D00)

This register is applicable to BCM5721, BCM5751, and BCM5752 only. This register is reset only by POR.

Table 572: Data Link Control Register (Offset 0x7D00)

Bit	*Field*	*Description*	*Init*	*Access*
31-25	Reserved	Write as 0, ignore when read.	0	-
24	PME Turn Off Message Handling Fix	Set this bit to 1 to enable the fix. Refer to E22_5751B1_11211 in the *5751-ES5xx-R Errata* for details.	0	R/W
23	Enable Data Link Layer Retry Logic Fix	Set to 1 to enable the fix. Refer to E13_5751B1_11121 in the *5751-ES5xx-R Errata* for details.	1 for BCM5721/ BCM5751 C0 and later chips, and otherwise 0.	R/W
22	Power Mgmt State Machine L0s Lockup Fix	Enable fix to transition out of L0s when link experiencing recovery. • 1 = Disable fix • 0 = Enable fix Refer to E6_5751B1_11080 in the *5751-ES5xx-R Errata* for details.	0	R/W
21	Enable Flow Control Credit Checking Fix	Enable this bit to check MPS or actually advertised credit. • 1 = Disable fix • 0 = Enable fix Refer to E5_5751B1_10674 in the *5751-ES5xx-R Errata* for details.	0	R/W
20	Enable L1 to L0 Transition when Device is Configured to D3 Hot	Enable this fix to transition back to L1 after waked up and D state is set at D3 state: • 1 = Disable fix • 0 = Enable fix Refer to E2_5751B1_10453 in the *5751-ES5xx-R Errata* for details.	0	R/W
19	Enable Pending Completion Packet Issue Fix	Enable this fix to wake up from L1 and flush out pending TLP. • 1 = Disable fix • 0 = Enable fix Refer to E1_5751B1_10452 in the *5751-ES5xx-R Errata* for details.	0	R/W
18	PLL REFSEL Switch Control	Enable this fix to allow PLL source clock to switch to local crystal at the absence of PCIe ref clock. • 1 = Enable switch • 0 = Disable switch Refer to E15_5751B1_11011 in the *5751-ES5xx-R Errata* for details.	1	R/W
17	Reserved	–	0	R/W
16	Power Management Control	Enable power management clock switching (allows core clk to be automatically muxed into PCIe clocks).	1	R/W

Table 572: Data Link Control Register (Offset 0x7D00) (Cont.)

Bit	Field	Description	Init	Access
15	Power Down SerDes Transmitter	Forces the SerDes transmitter into the low-power state (when cleared, the transmitter power state is controlled by the power management state machine).	0	R/W
14	Power Down SerDes PLL	Forces the SerDes PLL into the low-power state (when cleared, the PLL power state is controlled by the power management state machine).	0	R/W
13	Power Down SerDes Receiver	Forces the SerDes receiver into the low-power state (when cleared, receiver power state is controlled by power management state machine).	0	R/W
12	Enable Beacon	Enable transmission of In-band Beacon signal when waking system.	1	R/W
11	Automatic Timer Threshold Enable	• 1 = Enable automatic calculation of ACK Latency and Replay Timeout Values. • 0 = Use register values for ACK Latency and Replay Timeout.	1	R/W
10	Enable DLLP Timeout Mechanism	When set to 1, link is retrained if the DLLP receive timer expires without receiving a valid DLLP.	1	R/W
9	Check Receive Flow Control Credits	Check receive flow control credit consumption and report receive overflow errors when enabled.	1	R/W
8	Link Enable	Enable the data link layer functions.	1	R/W
7-0	Power Management Control	These bits enable automatic power management functions (power up/down or clock gating): • 7 = Enable Active State power management. • 6 = Enable PCI-PM power management (clearing this bit does not disable PM_PME message generation). • 5 = Enable SerDes transmitter power management. • 4 = Enable SerDes PLL power management. • 3 = Enable SerDes receiver power management. • 2 = Enable transaction layer power management (clock gating). • 1 = Enable data link layer power management (clock gating). • 0 = Enable physical layer power management (clock gating).	0xFF	R/W

DATA LINK STATUS REGISTER (OFFSET 0X7D04)

This register is applicable to BCM5721, BCM5751, and BCM5752 only.

Table 573: Data Link Status Register (Offset 0x7D04)

Bit	Field	Description	Init	Access
31-26	Reserved	Write as 0, ignore when read.	0	R/O
25-23	PHY Link State[1]	Current physical layer power state. • 000: L0 • 001: L0s • 010: L1 • 011: L2 • 100: others	100	R/O
22-19	Power Management State1	Current state of power management substate machine (see test doc for state mapping).	1000	R/O
18-17	Power Management Sub-State1	Current state of power management substate machine (see test doc for state mapping).	00	R/O
16	Data Link Up1	Data link is up (VC0 initialized).	0	R/O
15-10	Reserved	Write as 0, ignore when read.	0	R/O
9	Flow Control Update Timeout	Flow control update timeout error detected (DLLP receive timer expired without receiving valid DLLP).	0	RO/CR
8	Flow Control Receive Overflow	Flow control receive overflow error detected.	0	RO/CR
7	Flow Control Protocol Error	Flow control protocol error detected.	0	RO/CR
6	Data Link Protocol Error	Data link protocol error detected (pos or neg acknowledgement received with invalid TLP sequence number).	0	RO/CR
5	Replay Rollover	Replay counter rolled over (four consecutive retries without a positive acknowledgement received).	0	RO/CR
4	Replay Timeout	Replay timer expired (no ACK received within specified time).	0	RO/CR
3	NAK Received	Negative acknowledgement DLLP was received.	0	RO/CR
2	DLLP Error	Data link layer packet error detected.	0	RO/CR
1	Bad TLP Sequence Number	TLP received with invalid sequence number.	0	RO/CR
0	TLP Error	Transaction layer packet error detected (packet failed data link layer error checks).	0	RO/CR

[1] These bits are for debug only—they will always return 0 (except Data Link Up = 1) when read through the PCIe interface.

Data Link Attention Register (Offset 0x7D08)

This register is applicable to BCM5721, BCM5751, and BCM5752 only.

Table 574: Data Link Attention Register (Offset 0x7D08)

Bit	Field	Description	Init	Access
31-5	Reserved	Write as 0, ignore when read.	0	-
4	Data Link Layer Error Attention Indicator	Asserted when any of the following bits are set in the data link status register: • FC Update Timeout. • FC Receive Overflow. • FC Protocol Error. • Data Link Protocol Error. • Replay Rollover or Replay Timeout (read the "Data Link Status Register (Offset 0x7D04)" on page 587 to clear this bit).	0	R/O
3	NAK Received Counter Attention Indicator	Set when NAK received counter value is greater than or equal to attention threshold. Cleared when counter is read.	0	R/O
2	DLLP Error Counter Attention Indicator	Set when DLLP error counter value is greater than or equal to attention threshold. Cleared when counter is read.	0	R/O
1	TLP Bad Sequence Counter Attention Indicator	Set when TLP bad sequence counter value is greater than or equal to attention threshold. Cleared when counter is read.	0	R/O
0	TLP Error Counter Attention Indicator	Set when TLP error counter value is greater than or equal to attention threshold. Cleared when counter is read.	0	R/O

Data Link Attention Mask Register (Offset 0x7D0C)

This register is applicable to BCM5721, BCM5751, and BCM5752 only.

Table 575: Data Link Attention Mask Register (Offset 0x7D0C)

Bit	Field	Description	Init	Access
31-8	Reserved	Write as 0, ignore when read.	0	R/O
7-5	Attention Mask	Reserved for additional attention bits.	0	R/W
4	Data Link Layer Error Attention Mask	Data link error attention bit causes assertion of data link attention output when mask bit is set to 1.	0	R/W
3	NAK Received Counter Attention Mask	NAK received counter attention bit causes assertion of data link attention output when mask bit is set to 1.	0	R/W
2	DLLP Error Counter Attention Mask	DLLP error counter attention bit causes assertion of data link attention output when mask bit is set to 1.	0	R/W
1	TLP Bad Sequence Counter Attention Mask	TLP bad sequence counter attention bit causes assertion of data link attention output when mask bit is set to 1.	0	R/W
0	TLP Error Counter Attention Mask	TLP error counter attention bit causes assertion of data link attention output when mask bit is set to 1.	0	R/W

NEXT TRANSMIT SEQUENCE NUMBER DEBUG REGISTER (OFFSET 0X7D10)

This register is applicable to BCM5721, BCM5751, and BCM5752 only.

Table 576: Next Transmit Sequence Number Debug Register (Offset 0x7D10)

Bit	Field	Description	Init	Access
31-12	Reserved	Write as 0, ignore when read.	0	R/O
11-0	Next Transmit Sequence Number	Transmit sequence number for the next TLP to be sent.	0x000	R/W

ACKED TRANSMIT SEQUENCE NUMBER DEBUG REGISTER (OFFSET 0X7D14)

This register is applicable to BCM5721, BCM5751, and BCM5752 only.

Table 577: ACKed Transmit Sequence Number Debug Register (Offset 0x7D14)

Bit	Field	Description	Init	Access
31-12	Reserved	Write as 0, ignore when read.	0	R/O
11-0	ACKed Transmit Sequence Number	Sequence number for the last transmit TLP to be positively acknowledged.	0xFFF	R/W

PURGED TRANSMIT SEQUENCE NUMBER DEBUG REGISTER (OFFSET 0X7D18)

This register is applicable to BCM5721, BCM5751, and BCM5752 only.

Table 578: Purged Transmit Sequence Number Debug Register (Offset 0x7D18)

Bit	Field	Description	Init	Access
31-12	Reserved	Write as 0, ignore when read.	0	R/O
11-0	Purged Transmit Sequence Number	Sequence number for the last transmit TLP to be purged from retry buffer.	0xFFF	R/W

RECEIVE SEQUENCE NUMBER DEBUG REGISTER (OFFSET 0X7D1C)

This register is applicable to BCM5721, BCM5751, and BCM5752 only.

Table 579: Receive Sequence Number Debug Register (Offset 0x7D1C)

Bit	Field	Description	Init	Access
31-12	Reserved	Write as 0, ignore when read.	0	R/O
11-0	Receive Sequence Number	Receive sequence number for the last good TLP received.	0xFFF	R/W

DATA LINK REPLAY REGISTER (OFFSET 0X7D20)

This register is applicable to BCM5721, BCM5751, and BCM5752 only.

Table 580: Data Link Replay Register (Offset 0x7D20)

Bit	Field	Description	Init	Access
31-23	Reserved	Write as 0, ignore when read.	0	R/O
22-10	Replay Timeout Value	Replay timeout value in data link layer clock cycles (8 ns).	0x5cf	R/W
9-0	Retry Buffer Size	Physical size of retry buffer/16 bytes.	0xb0	R/W

DATA LINK ACK TIMEOUT REGISTER (OFFSET 0X7D24)

This register is applicable to BCM5721, BCM5751, and BCM5752 only.

Table 581: Data Link ACK Timeout Register (Offset 0x7D24)

Bit	Field	Description	Init	Access
31-11	Reserved	Write as 0, ignore when read.	0	R/O
10-0	ACK Latency Timeout Value	ACK latency timeout value in data link layer clock cycles (8 ns).	0xff	R/W

POWER MANAGEMENT THRESHOLD REGISTER (OFFSET 0X7D28)

This register is applicable to BCM5721, BCM5751, and BCM5752 only. This register is reset by either POR or GRC (soft) reset.

Table 582: Power Management Threshold Register (Offset 0x7D28)

Bit	Field	Description	Init	Access
31-24	Reserved	Write as 0, ignore when read.	0	R/O
23-20	D3C Re-enter Threshold	Minimum time in us before re-enter to L1 link state. **Note:** Refer to E1_5751B1_10452 in the *5751-ES5xx-R Errata* for more details.	• 0x9 for BCM5721 and BCM5751. • 0x8 for BCM5752.	R/W
19-16	D3C Exit Threshold	Minimum time in us that the link must stay in L1. **Note:** Refer to E1_5751B1_10452 in the *5751-ES5xx-R Errata* for more details.	0x2	R/W
15-8	L1 Threshold	Idle time before entering L1 low-power state (unit = 256 ns).	0x62	R/W
7-0	L0s Threshold	Idle time before entering L0s low-power state (unit = 16 ns).	0xfa	R/W

RETRY BUFFER WRITE POINTER DEBUG REGISTER (OFFSET 0X7D2C)

This register is applicable to BCM5721, BCM5751, and BCM5752 only.

Table 583: Retry Buffer Write Pointer Debug Register (Offset 0x7D2C)

Bit	Field	Description	Init	Access
31-11	Reserved	Write as 0, ignore when read.	0	R/O
10-0	Retry Buffer Write Pointer	Address of next DWORD to be written into retry buffer RAM.	0	R/W

RETRY BUFFER READ POINTER DEBUG REGISTER (OFFSET 0X7D30)

This register is applicable to BCM5721, BCM5751, and BCM5752 only.

Table 584: Retry Buffer Read Pointer Debug Register (Offset 0x7D30)

Bit	Field	Description	Init	Access
31-11	Reserved	Write as 0, ignore when read.	0	R/O
10-0	Retry Buffer Read Pointer	Address of next DWORD to be read from retry buffer RAM.	0	R/W

RETRY BUFFER PURGED POINTER DEBUG REGISTER (OFFSET 0X7D34)

This register is applicable to BCM5721, BCM5751, and BCM5752 only.

Table 585: Retry Buffer Purged Pointer Debug Register (Offset 0x7D34)

Bit	Field	Description	Init	Access
31-11	Reserved	Write as 0, ignore when read.	0	R/O
10-0	Retry Buffer Purged Pointer	Starting address of next TLP to be purged from retry buffer RAM.	0	R/W

RETRY BUFFER READ/WRITE DEBUG PORT (OFFSET 0X7D38)

This register is applicable to BCM5721, BCM5751, and BCM5752 only.

Table 586: Retry Buffer Read/Write Debug Port (Offset 0x7D38)

Bit	Field	Description	Init	Access
31-0	Retry Buffer Data	Data written to this address is written into the retry buffer RAM at the retry buffer write address. Reads to this address will return the data stored at the retry buffer read address in the retry buffer RAM.	–	R/W

Error Count Threshold Register (Offset 0x7D3C)

This register is applicable to BCM5721, BCM5751, and BCM5752 only.

Table 587: Error Count Threshold Register (Offset 0x7D3C)

Bit	Field	Description	Init	Access
31-15	Reserved	Write as 0, ignore when read.	0	R/O
14-12	Bad Sequence Number Count Threshold	Attention bits are set when error count reaches threshold. Threshold = 2^n.	0x7	R/W
11-8	NAK Received Count Threshold	Attention bits are set when error count reaches threshold. Threshold = 2^n.	0xF	R/W
7-4	DLLP Error Count Threshold	Attention bits are set when error count reaches threshold. Threshold = 2^n.	0xF	R/W
3-0	TLP Error Count Threshold	Attention bits are set when error count reaches threshold. Threshold = 2^n.	0xF	R/W

TLP Error Counter Register (Offset 0x7D40)

This register is applicable to BCM5721, BCM5751, and BCM5752 only.

Table 588: TLP Error Counter Register (Offset 0x7D40)

Bit	Field	Description	Init	Access
31-24	Reserved	Write as 0, ignore when read.	0	R/O
23-16	TLP Bad Sequence Number Counter	Counts number of TLPs with bad sequence number received since last read. Counter freezes at max value and will be cleared to one if event occurs simultaneously to read.	0	RO/CR
15-0	TLP Error Counter	Counts number of bad TLPs received (includes bad LCRC, bad length or bad sequence number) since last read. Counter freezes at max value and will be cleared to one if event occurs simultaneously to read.	0	RO/CR

DLLP Error Counter (Offset 0x7D44)

This register is applicable to BCM5721, BCM5751, and BCM5752 only.

Table 589: DLLP Error Counter (Offset 0x7D44)

Bit	Field	Description	Init	Access
31-16	Reserved	Write as 0, ignore when read.	0	R/O
15-0	DLLP Error Counter	Counts number of bad DLLPs received (includes bad LCRC or bad length) since last read. Counter freezes at max value and will be cleared to one if event occurs simultaneously to read.	0	RO/CR

NAK Received Counter (Offset 0x7D48)

This register is applicable to BCM5721, BCM5751, and BCM5752 only.

Table 590: NAK Received Counter (Offset 0x7D48)

Bit	Field	Description	Init	Access
31-16	Reserved	Write as 0, ignore when read.	0	R/O
15-0	NAK Received Counter	Counts number of NAK DLLPs received since last read. Counter freezes at max value and will be cleared to one if event occurs simultaneously to counter read.	0	RO/CR

Data Link Test Register (Offset 0x7D4C)

This register is applicable to BCM5721, BCM5751, and BCM5752 only.

Table 591: Data Link Test Register (Offset 0x7D4C)

Bit	Field	Description	Init	Access
31-16	Reserved	Write as 0, ignore when read.	0	R/O
15	Store Receive TLPs	Write received TLPs into retry buffer instead of transmitted TLPs.	0	R/W
14	Disable TLPs	Disable transmission of TLPs.	0	R/W
13	Disable DLLPs	Disable transmission and reception of DLLPs.	0	R/W
12	Force PHY Link Up	Force PHY link input to data link layer to be up.	0	R/W
11	Bypass Flow Control	Force flow control init flags to be set and available TX flow control credits to infinite.	0	R/W
10	Enable RAM Core Clock Margin Test Mode	Enable retry buffer RAM core clock margin test mode.	0	R/W
9	Enable RAM Overstress Test Mode	Enable retry buffer RAM overstress test mode.	0	R/W
8	Enable RAM Read Margin Test Mode	Enable retry buffer RAM read margin test mode.	0	R/W
7	Speed up Completion Timer	Speed up completion timer and LED blink rate for simulation.	0	R/W
6	Speed up Replay Timer	Speed up replay timer for simulation.	0	R/W
5	Speed up ACK Latency Timer	Speed up ACK latency timer for simulation.	0	R/W
4	Speed up PME Service Timer	Speed up PME service timer for simulation.	0	R/W
3	Force Purge	Purge the contents of the retry buffer.	0	W/SC
2	Force Retry	Retransmit the contents of the retry buffer.	0	W/SC
1	Invert CRC	Force entire LCRC to be inverted.	0	R/W
0	Send Bad CRC	Force last bit of LCRC to be inverted.	0	R/W

PACKET BIST REGISTER (OFFSET 0X7D50)

This register is applicable to BCM5721, BCM5751, and BCM5752 only.

Table 592: Packet BIST Register (Offset 0x7D50)

Bit	Field	Description	Init	Access
31-24	Reserved	Write as 0, ignore when read.	0	R/O
23	Packet Checker Locked	Packet checker has locked to received data sequence.	0	R/O
22	Receive Mismatch	Receive data or packet length did not match pseudo-random sequence. This bit sticks high and can only be cleared by disabling the packet generator test mode or clearing the Transmit Start bit.	0	R/O
21	Enable Random TLP Length	• 1 = Transmit random length TLPs. • 0 = Transmit fixed length TLPs.	1	R/W
20-10	TLP Length	Transmit TLP length is equal to this field + 3 DWORDS. When sending random length TLPs, this field is ANDed with the random generator output in order to limit the maximum length.	0x1FF	R/W
9	Enable Random IPG Length	• 1 = Transmit random length IPGs. • 0 = Transmit fixed length IPGs.	1	R/W
8-2	IPG Length	Transmit IPG length is equal to this field + two DWORDS. When sending random length IPGs, this field is ANDed with the random generator output in order to limit the maximum length.	0x1F	R/W
1	Transmit Start	Start transmitting TLPs. TLP transmission will be halted when this bit is cleared or when error condition occurs (receive data mismatch, DLLP error or TLP error).	1	R/W
0	Enable Packet Generator Test Mode	Transmit continuous stream of random or fixed length TLPs containing pseudorandom data, separated by random or fixed length IPGs. If TLPs are looped back, received TLPs are checked vs. expected length and data content. Received TLPs will be passed through retry buffer if the Store Receive TLPs bit is set in the test register.	0	R/W

PHY MODE REGISTER (OFFSET 0X7E00)

This register is applicable to BCM5721, BCM5751, and BCM5752 only.

Table 593: PHY Mode Register (Offset 0x7E00)

Bit	Field	Description	Init	Access
31-2	Reserved	Write as 0, ignore when read.	0	R/O
1	Link disable	Disable the logical PHY layer functions.	0	R/W
0	Soft reset	Softreset to the phylogical block. This bit will be self-cleared after four clock cycles.	0	R/W

PHY/Link Status Register (Offset 0x7E04)

This register is applicable to BCM5721, BCM5751, and BCM5752 only.

Table 594: PHY/Link Status Register (Offset 0x7E04)

Bit	Field	Description	Init	Access
31-8	Reserved	-	0	R/O
7	Link partner request loopback	Link partner requested remote loopback mode during training process.	0	R/O
6	Link partner disable scrambler	The link partner disabled the scrambler during training process.	0	R/O
5	Extended Synch	Extended synchronization from PCI configuration register. If set, 4K FTS ordered sets must be sent during link recovery.	0	R/O
4	Polarity inverted	Lane polarity is inverted.	0	R/O
3	Link Up	The link training process is completed and link is ready for use.	0	R/O
2	Link training	The link is in the training process.	0	R/O
1	Receive data valid	Symbol synchronization is achieved and receive data is valid	0	R/O
0		-		

PHY/Link LTSSM Control Register (Offset 0x7E08)

This register is applicable to BCM5721, BCM5751, and BCM5752 only.

Table 595: PHY/Link LTSSM Control Register (Offset 0x7E08)

Bit	Field	Description	Init	Access
31-8	Reserved	-	0	R/O
7	DisableScramble	Disable scrambling and de-scrambling.	0	R/W
6	DetectState	High layer directs LTSSM to detect state if set. The bit is cleared when LTSSM entered into detect state.	0	R/W
5	PollingState	High layer directs LTSSM to Polling state if set. The bit is cleared when LTSSM entered into Polling state.	0	R/W
4	ConfigState	High layer directs LTSSM to configuration state if set. The bit is cleared when LTSSM entered into configuration state.	0	R/W
3	RecovState	High layer directs LTSSM to recovery state if set. The bit is cleared when LTSSM entered into recovery state.	0	R/W
2	ExtLBState	High layer directs LTSSM to external loopback master state if set. The bit is cleared when LTSSM entered into master external loopback state.	0	R/W
1	ResetState	High layer directs LTSSM to hot reset state if set. The bit is cleared when LTSSM exited out of the hot reset state.	0	R/W
0	DisableState	High layer directs LTSSM to disable state if set. The bit is cleared when LTSSM entered into disable state.	0	R/W

PHY/LINK TRAINING LINK NUMBER (OFFSET 0X7E0C)

This register is applicable to BCM5721, BCM5751, and BCM5752 only.

Table 596: PHY/Link Training Link Number (Offset 0x7E0C)

Bit	Field	Description	Init	Access
31-8	Reserved	-	0	R/O
7-0	Lane Number	Lane Number within component	PAD	R/O

PHY/LINK TRAINING LANE NUMBER (OFFSET 0X7E10)

This register is applicable to BCM5721, BCM5751, and BCM5752 only.

Table 597: PHY/Link Training Lane Number (Offset 0x7E10)

Bit	Field	Description	Init	Access
31-8	Reserved	-	0	R/O
7-0	Lane Number	Lane Number within link	PAD	R/O

PHY/LINK TRAINING N_FTS (OFFSET 0X7E14)

This register is applicable to BCM5721, BCM5751, and BCM5752 only.

Table 598: PHY/Link Training N_FTS (Offset 0x7E14)

Bit	Field	Description	Init	Access
31-16	Reserved	-	0	R/O
15-8	Inbound N_FTS	Inbound Maximum number of FTS ordered sets to be sent when transitioning from L0s to L0 to achieve bit and framing synchronization.	0xFF	R/O
7-0	Outbound N_FTS	Outbound Maximum number of FTS ordered sets to be sent when transitioning from L0s to L0 to achieve bit and framing synchronization.	0xFF	R/W

PHY ATTENTION REGISTER (OFFSET 0X7E18)

This register is applicable to BCM5721, BCM5751, and BCM5752 only.

Table 599: PHY Attention Register (Offset 0x7E18)

Bit	Field	Description	Init	Access
31-8	Reserved	-	0	R/O
7	Hot reset	Hot reset event. Set by hot reset and cleared by explicitly writing 1.	0	W2C
6	Link down	Link down event. When link status transitions from up to down, this event bit will be set.	0	W2C
5	Training error	LTSSM training error.	0	W2C
4	Buffer overrun	Receive elastic buffer overrun.	0	W2C
3	Buffer underrun	Receive elastic buffer underrun.	0	W2C
2	Receive framing error	Receive framing error. Set when receive framing error count exceeds its threshold.	0	W2C
1	Receive disparity error	Receive 8b/10b running disparity error. Set when 8b10b disparity count exceeds its threshold.	0	W2C
0	Receive code error	Receive 8b/10b code error. Set when 8b/10b error count exceeds its threshold.	0	W2C

PHY ATTENTION MASK REGISTER (OFFSET 0X7E1C)

This register is applicable to BCM5721, BCM5751, and BCM5752 only.

Table 600: PHY Attention Mask Register (Offset 0x7E1C)

Bit	Field	Description	Init	Access
31-8	Reserved	-	0	R/O
7	Hot reset mask	Hot reset event mask bit.	0	R/W
6	Link down mask	Link down event mask bit.	0	R/W
5	Training error mask	LTSSM training error mask bit.	0	R/W
4	Buffer overrun mask	Receive elastic buffer overrun mask bit.	0	R/W
3	Buffer underrun mask	Receive elastic buffer underrun mask bit.	0	R/W
2	Receive frame error mask	Receive frame error mask bit.	0	R/W
1	Receive disparity error mask	Receive 8b/10b running disparity error mask bit.	0	R/W
0	Receive code error mask	Receive 8b/10b code error mask bit.	0	R/W

PHY Receive Error Counter (Offset 0x7E20)

This register is applicable to BCM5721, BCM5751, and BCM5752 only.

Table 601: PHY Receive Error Counter (Offset 0x7E20)

Bit	Field	Description	Init	Access
31-16	Disparity error count	Receive 8b/10b running disparity error count.	0	R2C
15-0	Code error count	Receive 8b/10b coding error count.	0	R2C

PHY Receive Framing Error Counter (Offset 0x7E24)

This register is applicable to BCM5721, BCM5751, and BCM5752 only.

Table 602: PHY Receive Framing Error Counter (Offset 0x7E24)

Bit	Field	Description	Init	Access
31-16	Reserved	-	0	R/O
15-0	Framing error count	Receive framing error count.	0	R2C

PHY Receive Error Threshold Register (Offset 0x7E28)

This register is applicable to BCM5721, BCM5751, and BCM5752 only.

Table 603: PHY Receive Error Threshold Register (Offset 0x7E28)

Bit	Field	Description	Init	Access
31-12	Reserved	-	0	R/O
11-8	Frame Error Threshold	Receive frame error threshold. When the frame error count exceeds this threshold. The frame error attention bit is set. Threshold=2^n, where n=bits(11:8).	0xF	R/W
7-4	Disparity Error Threshold	Receive 8b10b running disparity error threshold. When the running disparity error count exceed this threshold, the disparity error will be set. Threshold = 2^n, where n = bits(7:4).	0xF	R/W
3-0	Code Error Threshold	Receive 8b10b coding error threshold. When the code error count exceeds threshold, the code error attention bit is set. Threshold = 2^n, where n = bits(3:0).	0xF	R/W

PHY Test Control Register (Offset 0x7E2C)

This register is applicable to BCM5721, BCM5751, and BCM5752 only.

Table 604: PHY Test Control Register (Offset 0x7E2C)

Bit	Field	Description	Init	Access
31-12	Reserved	-	0	R/O
11	Enable Immediate L1 Exit Issue	Enable the immediate L1 Exit Fix • 0 = Disable Fix • 1 = Enable Fix ***Note:*** Refer to E1_5751B0_09901 in the *5751-ES4xx-R Errata* for details.	0	R/W
10	Reserved	-	0	R/W
9	Enable x16 Slot L1 Entry Fix	Enable the x16 Slot L1 Entry problem Fix • 0 = Disable Fix • 1 = Enable Fix Enable this bit to avoid possible premature exit from the L1 state to recovery state.	0	R/W
	Reserved (BCM5752 only)	Should not be written with a value other than default value read from this bit.	0	R/W
8	Enable Electrical Ordered Set Not Detected Fix	Enable the Electrical Ordered Set not Detected Fix • 0 = Disable Fix • 1 = Enable Fix The BCM5751/BCM5721 receiver fails to detect an electrical idle ordered set if it is sent after a partially completed DLLP or TLP. This causes the link training state machine to see an unexpected electrical idle and transition to the recovery state instead of the L0s, L1, or L2 state. Set this bit to 1 to fix the above problem.	0	R/W
	Reserved (BCM5752 only)	Should not be written with a value other than default value read from this bit.	0	R/W
7	Reserved	Should not be written with a value other than default value read from this bit.	0	R/W
6	PCIe 1.0 Mode	When this bit is: • Set to 1, the physical layer LTSSM state machine operates in PCIe 1.0 mode. • Clear, it operates in PCIe 1.0a mode.	0	R/W
5	PCIe 1.0 Scrambler	When this bit is: • Set to 1, the PCIe scrambler operates in PCIe 1.0 mode. • Clear, it operates in PCIe 1.0a mode.	0	R/W
4	Fast Symbol Lock Up	When this bit is: • Set to 1, the symbol boundary locks after receiving the first COM symbol. • Clear, the symbol boundary locks after receiving four COM symbols within a 64-symbol time.	0	R/W
3	Down Stream Lane	Set this bit to change the link to be a downstream lane (or upstream component).	0	R/W
2	Training Bypass	Set to bypass link initialization and configuration process.	0	R/W

Table 604: PHY Test Control Register (Offset 0x7E2C) (Cont.)

Bit	Field	Description	Init	Access
1	External Loopback	Force remote (external) loopback test mode.	0	R/W
0	Internal Loopback	Force internal parallel loopback test mode.	0	R/W

PHY/SerDes Control Override Register (Offset 0x7E30)

This register is applicable to BCM5721, BCM5751, and BCM5752 only.

Table 605: PHY/SerDes Control Override Register (Offset 0x7E30)

Bit	Field	Description	Init	Access
31-18	Reserved		0	R/O
17	obsvElecIdleValue	Override value for the obsvElecIdle signal from SerDes.	0	R/W
16	obsvElecIdleOverride	Set to override the obsvElecIdle signal value from SerDes with the value in bit 17 of this register.	0	R/W
15	pllIsUpValue	Override value for pllIsUp signal form SerDes.	0	R/W
14	pllIsUpOverride	Set to override the pllisUp signal value from SerDes with the value in bit 15 of this register.	0	R/W
13	rcvrDetValue	Override value for rcvrDetected signal from SerDes.	0	R/W
12	rcvrDetOverride	Set to override the rcvrDetected signal from SerDes with the value in bit 13 of this register.	0	R/W
11-10	rcvrDetTimeControl	Time unit of the rcvrDetectionTime: • 2’b00: Symbol time. • 2’b01: 64 ns. • 2’b10: 1 µs.	0x0	R/W
9-0	rcvrDetectionTime	Time value that the PHY logical layer uses for timing receiver detection sequences.	0x3FF	R/W

PHY Timing Parameter Override Register (Offset 0x7E34)

This register is applicable to BCM5721, BCM5751, and BCM5752 only.

Table 606: PHY Timing Parameter Override Register (Offset 0x7E34)

Bit	Field	Description	Init	Access
31	ts1NumOverride	Set to override the TS1 number to be sen out in polling active state with the value in bit (27:16) of this register from the specification defined value.	0	R/W
30	txIdleMinOverride	Set to override the min time for a transmitter to stay with the value in bit (15:8) of this register from the specification defined value.	0	R/W
29	txIdle2IdleOverride	Set to override the Max time for electrical idle transition with the value in bit (7:0) of this register from the specification defined value.	0	R/W
28	Reserved	-	0	R/O
27-16	N_TS1InPollingActive	TS1 number needed to be sent in Polling active state.	0x400	R/W
15-8	txIdleMinTime	Minimum time (in symbol time) a transmitter must be in electrical idle.	0x5	R/W
7-0	txIdleSettoIdleTime	Maximum time (in symbol time) to transition to a valid electrical idle after sending an electrical idle ordered-set.	0x2	R/W

PHY Hardware Diagnostic 1 Register (Offset 0x7E38)

This register is applicable to BCM5721, BCM5751, and BCM5752 only.

Table 607: PHY Hardware Diagnostic 1 Register (Offset 0x7E38)

Bit	Field	Description	Init	Access
31-10	Reserved	-	0	R/O
9-4	Transmit State Machine State	Transmit state machine states: • 9:8: TX Data State • 7:4: TX Main State	0	R/O
3-0	Receive State Machine State	Receive state machine states: • 3:0: RX Main State	0	R/O

PHY HARDWARE DIAGNOSTIC 2 REGISTER (OFFSET 0X7E3C)

This register is applicable to BCM5721, BCM5751, and BCM5752 only.

Table 608: PHY Hardware Diagnostic 2 Register (Offset 0x7E3C)

Bit	Field	Description	Init	Access
31-0	LTSSM State Machine State	LTSSM state machine states: • 31-28: Main State. • 27-26: Detect substate. • 25-23: Polling substate. • 22-20: Configuration substate. • 19-18: Recover substate. • 17-16: RX L0s substate. • 15-14: RX L0s substate. • 13-12: L1 substate. • 11-10: L2 substate. • 9-8: Disable substate. • 7-6: Loopback substate. • 5-4: Reset substate. • 3-0: Reserved.	0	R/O

TRANSCEIVER REGISTERS

The BCM5701 and later MAC controllers have an integrated 10/100/1000T PHY or SerDes PHY or 1000BASE-X PHY. This section describes the MII registers of the integrated 10/100/1000T and 1000BASE-X PHY transceivers. The access to the transceiver registers is provided indirectly through the MII Communication Register (see "MI Communication Register (Offset 0x44C)" on page 388) of the MAC. The transceiver registers are accessed with the PHY_Addr bit of the MII Communication Register set to 0x1. The integrated transceiver contains the set of registers shown in Table 609.

There is no integrated PHY in BCM5700 and hence the transceiver registers in this section are not applicable to the BCM5700 MAC. The BCM5703S and BCM5704S support the Fiber Media through an integrated SerDes PHY whose control registers are specified in the MAC registers. So the transceiver registers described in this section are also not applicable to BCM5703S and BCM5704S devices.

The BCM5714S and BCM5715S devices with integrated 1000BASE-X PHY also support the MII registers described in this section. The MII registers at offset 0x00-0x0F of 1000BASE-X PHY are different from the MII registers 0x00-0x0F of 10/100/1000T PHY. This section also covers the description of 1000BASE-X MII registers which are applicable to BCM5714S and BCM5715S devices.

Table 609: Transceiver Register Map

Reg_Addr	***Register***
00h	MII Control register
01h	MII Status register
02h	PHY Identifier
03h	PHY Identifier
04h	Auto-negotiation Advertisement
05h	Auto-negotiation Link Partner BASE Page Ability
06h	Auto-negotiation Expansion register
07h	Auto-negotiation Next Page Transmit
08h	Auto-negotiation Link Partner Received Next Page
09h	1000BASE-T Control register
0Ah	1000BASE-T Status register
0B-0Eh	Reserved*
0Fh	IEEE Extended Status register
10h	PHY Extended Control register
11h	PHY Extended Status register
12h	Receive Error Counter
13h	False Carrier Sense Counter
14h	Receiver NOT_OK Counters
15h-17h	Reserved*
18h	Auxiliary Control register
19h	Auxiliary Status Summary register
1Ah	Interrupt Status register
1Bh	Interrupt Mask register

Table 609: Transceiver Register Map (Cont.)

Reg_Addr	Register
1Ch	Reserved*
1D-1Fh	Test Registers

* Reserved registers should never be read or written.

00-0FH 1000BT/100BTX/10BT REGISTERS

MII Control Register (PHY_Addr = 0x1, Reg_Addr = 00h)

Table 610: MII Control Register (PHY_Addr = 0x1, Reg_Addr = 00h)

Bit	Field	Description	Init	Access
15	Reset	To reset the transceiver by software control, a 1 can be sent to bit 15 of the MII Control Register using an MII write operation. The bit clears itself after the reset process is complete and need not 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. When this bit is read, it returns a one during the reset process, otherwise, it returns a 0. • 1 = PHY reset • 0 = Normal operation	0	R/W SC
14	Internal Loopback	The BCM57XX can be placed into internal loopback mode by writing a 1 to 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. When this bit is read, it returns a 1 when the chip is in loopback mode, otherwise, it returns a 0. • 1 = Loopback mode • 0 = Normal operation	0	R/W
13	Speed Selection (LSB)	Bits 6 and 13 of the MII Control Register can be used to manually select the speed of operation when auto-negotiation is disabled. When bit 6 is set and bit 13 is cleared, 1000BASE-T is selected. When bit 6 is cleared and bit 13 is set, 100BASE-T is selected. When both bits are cleared, 10BASE-T is selected. Other configurations are reserved and cause unpredictable behavior in the BCM57XX. When read, these bits return the last value written. Bits 6,13: • 1 1 = Reserved • 1 0 = 1000 Mbps • 0 1 = 100 Mbps • 0 0 = 10 Mbps	0	R/W

Table 610: MII Control Register (PHY_Addr = 0x1, Reg_Addr = 00h) (Cont.)

Bit	Field	Description	Init	Access
12	Auto-negotiation Enable	When bit 12 of the MII Control Register is set, the BCM57XX mode of operation is controlled by auto-negotiation. When this bit is cleared, the BCM57XX mode of operation is determined by the manual speed, duplex mode, and master/slave configuration bits. When this bit is read, it returns a 1 when auto-negotiation is enabled, otherwise, it returns a 0. • 1 = Auto-negotiation enabled • 0 = Auto-negotiation disabled	1	R/W
11	Power Down	When bit 11 of the MII Control Register is written to 1, the transceiver is placed into low-power standby mode. • 1 = Power dow • 0 = Normal operation.	0	R/W
10	Isolate	• 1 = Electrically isolate PHY from MII • 0 = Normal operation	0	R/W
9	Restart Auto-negotiation	Bit 9 of the MII Control Register allows the auto-negotiation process to be restarted, regardless of the current state of the auto-negotiation state machine. When auto-negotiation is enabled, writing a 1 to this bit restarts the auto-negotiation process. Writing a 0 to this bit has no effect. This bit is self-clearing, so it always returns 0 when read. • 1 = Restarting auto-negotiation • 0 = Auto-negotiation restart complete	0	R/W SC
8	Duplex Mode	When auto-negotiation is disabled, the duplex mode of the BCM57XX can be controlled by writing to bit 8 of the Control Register. Setting this bit to a 1 forces the BCM57XX into full-duplex operation, and setting this bit to a 0 forces the BCM57XX into half-duplex operation. • 1 = Full-duplex • 0 = Half-duplex	1	R/W
7	Collision Test Enable	• 1 = Enable the collision test mode • 0 = Disable the collision test mode	0	R/W
6	Speed Selection (MSB)	See bit 13.	1	R/W
5:0	Reserved	Write as 0, ignore on read.	-	R/W

MII Status Register (PHY_Addr = 0x1, Reg_Addr = 01h)

Table 611: MII Status Register (PHY_Addr = 0x1, Reg_Addr = 01h)

Bit	Field	Description	Init	Access
15	100BASE-T4 Capable	The BCM57XX is not capable of 100BASE-T4 operation, and returns a 0 when bit 15 of the MII Status register is read. • 1 = 100BASE-T4 capable • 0 = Not 100BASE-T4 capable	0	R/O L
14	100BASE-X Full-Duplex Capable	The BCM57XX is capable of 100BASE-TX full-duplex operation, and returns a 1 when bit 14 of the MII Status register is read. • 1 = 100BASE-X full-duplex capable • 0 = Not 100BASE-X full-duplex capable	1	R/O H
13	100BASE-X Half-Duplex Capable	The BCM57XX is capable of 100BASE-X half-duplex operation, and returns a 1 when bit 13 of the MII Status register is read. • 1 = 100BASE-X half-duplex capable • 0 = Not 100BASE-X half-duplex capable	1	R/O H
12	10BASE-T Full-Duplex Capable	The BCM57XX is capable of 10BASE-T full-duplex operation, and returns a 1 when bit 12 of the MII Status register is read. • 1 = 10BASE-T full-duplex capable • 0 = Not 10BASE-T full-duplex capable	1	R/O H
11	10BASE-T Half-Duplex Capable	The BCM57XX is capable of 10BASE-T half-duplex operation, and returns a 1 when bit 11 of the MII Status register is read. • 1 = 10BASE-T half-duplex capable • 0 = Not 10BASE-T half-duplex capable.	1	R/O H
10	100BASE-T2 Full-Duplex Capable	The BCM57XX is not capable of 100BASE-T2 full-duplex operation, and returns a 0 when bit 10 of the MII Status register is read. • 1 = 100BASE-T2 full-duplex capable • 0 = Not 100BASE-T2 full-duplex capable	0	R/O L
9	100BASE-T2 Half-Duplex Capable	The BCM57XX is not capable of 100BASE-T2 half-duplex operation, and returns a 0 when bit 9 of the MII Status register is read. • 1 = 100BASE-T2 half-duplex capable • 0 = Not 100BASE-T2 half-duplex capable.	0	R/O L
8	Extended Status	The BCM57XX contains the extended status register at Reg_Addr = 0Fh, and returns a 1 when bit 8 of the MII Status register is read. • 1 = Extended status information in reg 0Fh • 0 = No extended status information in reg 0Fh	1	R/O H
7	Reserved	Ignore on read	-	R/O

Table 611: MII Status Register (PHY_Addr = 0x1, Reg_Addr = 01h) (Cont.)

Bit	Field	Description	Init	Access
6	Management Frames Preamble Suppression	The BCM57XX accepts MII management frames whether or not they are preceded by the preamble pattern, and returns a 1 when bit 6 of the status register is read. The preamble is still required on the first read or write. • 1 = Preamble can be suppressed • 0 = Preamble always required	1	R/O H
5	Auto-negotiation Complete	The BCM57XX returns a 1 on bit 15 of the Auxiliary Status Summary Register (see "Auxiliary Status Summary Register (PHY_Addr = 0x1, Reg_Addr = 19h)" on page 657) when auto-negotiation has completed and the contents of registers 4, 5, and 6 are valid. This bit returns a 0 while auto-negotiation is in progress. • 1 = Auto-negotiation complete • 0 = Auto-negotiation in progress	0	R/O
4	Remote Fault	The BCM57XX returns a 1 on bit 4 of the MII Status register when its link partner has signalled a remote fault condition. When a remote fault occurs, the bit is latched at 1 and remains so until the register is read, and the remote fault condition has been cleared. • 1 = Remote fault detected • 0 = No remote fault detected	0	R/O LH
3	Auto-negotiation Ability	The BCM57XX is capable of performing IEEE auto-negotiation, and returns a 1 when bit 3 of the MII Status register is read, regardless of whether the auto-negotiation function has been disabled. • 1 = Auto-negotiation capable • 0 = Not auto-negotiation capable	1	R/O H
2	Link Status	The BCM57XX returns a 1 on bit 2 of the MII Status Register 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 BCM57XX is in the Link Pass state. • 1 = Link is up (Link Pass state) • 0 = Link is down (Link Fail state)	0	R/O LL
1	Jabber Detect	Jabber detection is performed within the PHY and the result is latched into this bit. The BCM57XX returns a 1 in bit 1 of the status register when a jabber condition has been detected. The bit is cleared by the read. • 1 = Jabber condition detected • 0 = No jabber condition detected	0	R/O LH
0	Extended Capability	The BCM57XX supports extended capability registers, and returns a 1 when bit 0 of the MII Status register is read. • 1 = Extended register capabilities • 0 = No extended register capabilities	1	R/O H

PHY Identifier Registers (PHY_Addr = 0x1, Reg_Addresses 02h and 03h)

Table 612: PHY Identifier Registers (PHY_Addr = 0x1, Reg_Addresses 02h and 03h)

Bit	Field	Description	Init	Access
15:0	Address = 02: ID MSBs	16 MSBs of PHY Identifier	0x0020	R/O
15:0	Address = 03: ID LSBs (BCM5703C and BCM5703S only)	16 LSBs of PHY Identifier	0x616[1]	R/O
	Address = 03: ID LSBs (BCM5705 only)	16 LSBs of PHY Identifier	0x61[1]	R/O
	Address = 03: ID LSBs (BCM5704C and BCM5704S only)	16 LSBs of PHY Identifier	0x619	-
	Address = 03: ID LSBs (BCM5721, and BCM5751 only)	16 LSBs of PHY Identifier	0x618[1]	R/O

[1] The revision number (*n*) changes with each silicon revision.

Auto-Negotiation Advertisement Register (PHY_Addr = 0x1, Reg_Addr = 04h)

Table 613: Auto-Negotiation Advertisement Register (PHY_Addr = 0x1, Reg_Addr = 04h)

Bit	Field	Description	Init	Access
15	Next Page	Bit 15 of the auto-negotiation advertisement register must be written to 1 when the management software wishes to control Next Page exchange. When this bit is written to 0, Next Page exchange is controlled automatically by the BCM57XX. When this bit is 0 and the BCM57XX is advertising no 1000BASE-T capability, no exchange of Next Pages occurs. • 1 = Next Page ability supported • 0 = Next Page ability not supported	0	R/W
14	Reserved	Write as 0, ignore on read.	0	R/O
13	Remote Fault	Writing a 1 to bit 13 of the auto-negotiation advertisement register 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. • 1 = Advertise remote fault detected • 0 = Advertise no remote fault detected	0	R/W
12	Reserved Technologies	Bit 12 of the auto-negotiation advertisement register is reserved for future versions of the auto-negotiation standard and must always be written as 0. Write as 0, ignore on read.	0	R/W

Table 613: Auto-Negotiation Advertisement Register (PHY_Addr = 0x1, Reg_Addr = 04h) (Cont.)

Bit	Field	Description	Init	Access
11	Asymmetric Pause	When Bit 11 of the auto-negotiation advertisement register is written to 1, the BCM57XX advertises that asymmetric pause is desired. When the bit is written to 0, the BCM57XX 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 of the auto-negotiation advertisement register indicates the desired direction of pause operation. Setting bit 10 to 1 indicates that pause frames flow toward the BCM57XX. Setting bit 10 to 0 indicates that pause frames flow toward the link partner. • 1 = Advertise asymmetric pause • 0 = Advertise no asymmetric pause	0	R/W
10	Pause Capable	When Bit 10 of the auto-negotiation advertisement register is written to 1, the BCM57XX advertises full-duplex pause capability. When the bit is written to 0, the BCM57XX advertises no pause capability. This bit returns a 1 when advertising pause capability, otherwise, it returns a 0. • 1 = Capable of full-duplex Pause operation • 0 = Not capable of Pause operation	0	R/W
9	100BASE-T4 Capability	When bit 9 of the auto-negotiation advertisement register is written to 1, the BCM57XX advertises 100BASE-T4 capability. When the bit is written to 0, the BCM57XX advertises no 100BASE-T4 capability. This bit returns a 1 when advertising 100BASE-T4 capability, otherwise, it returns a 0. • 1 = 100BASE-T4 capable • 0 = Not 100BASE-T4 capable	0	R/W
8	100BASE-TX Full-Duplex Capability	When bit 8 of the auto-negotiation advertisement register is written to 1, the BCM57XX advertises 100BASE-TX full-duplex capability. When the bit is written to 0, the BCM57XX advertises no 100BASE-TX full-duplex capability. This bit returns a 1 when advertising 100BASE-TX full-duplex capability, otherwise, it returns a 0. • 1 = 100BASE-TX full-duplex capable • 0 = Not 100BASE-TX full-duplex capable	1	R/W
7	100BASE-TX Half-Duplex Capability	When bit 7 of the auto-negotiation advertisement register is written to 1, the BCM57XX advertises 100BASE-TX half-duplex capability. When the bit is written to 0, the BCM57XX advertises no 100BASE-TX half-duplex capability. This bit returns a 1 when advertising 100BASE-TX half-duplex capability, otherwise, it returns a 0. • 1 = 100BASE-TX half-duplex capable • 0 = Not 100BASE-TX half-duplex capable	1	R/W
6	10BASE-T Full-Duplex Capability	When bit 6 of the auto-negotiation advertisement register is written to 1, the BCM57XX advertises 10BASE-T full-duplex capability. When the bit is written to 0, the BCM57XX advertises no 10BASE-T full-duplex capability. This bit returns a 1 when advertising 10BASE-T full-duplex capability, otherwise, it returns a 0. • 1 = 10BASE-T full-duplex capable • 0 = Not 10BASE-T full-duplex capable	1	R/W

Table 613: Auto-Negotiation Advertisement Register (PHY_Addr = 0x1, Reg_Addr = 04h) (Cont.)

Bit	Field	Description	Init	Access
5	10BASE-T Half-Duplex Capability	When bit 5 of the auto-negotiation advertisement register is written to 1, the BCM57XX advertises 10BASE-T half-duplex capability. When the bit is written to 0, the BCM57XX advertises no 10BASE-T half-duplex capability. This bit returns a 1 when advertising 10BASE-T half-duplex capability, otherwise, it returns a 0. • 1 = 10BASE-T half-duplex capable • 0 = Not 10BASE-T half-duplex capable	1	R/W
4:0	Selector Field	Bits 4:0 of the auto-negotiation advertisement register indicate the protocol type. The value 00001 indicates that the BCM57XX belongs to the 802.3 class of PHY transceivers. 00001 indicates IEEE 802.3 CSMA/CD	00001	R/W

Auto-Negotiation Link Partner Ability Register (PHY_Addr = 0x1, Reg_Addr = 05h)

This register is addressed through the MII Communication Register (see "MI Communication Register (Offset 0x44C)" on page 388) with the PHY Address field = 0x1 and the Register Address field = 05h.

This register is not applicable to the BCM5700 MAC.

The values contained in the Auto-negotiation Link Partner Ability Register are only guaranteed to be valid once auto-negotiation has successfully completed, as indicated by bit 5 of the MII status register.

Table 614: Auto-Negotiation Link Partner Ability Register (PHY_Addr = 0x1, Reg_Addr = 05h)

Bit	Field	Description	Init	Access
15	Next Page	The BCM57XX returns a 1 on bit 15 of the Link Partner Ability Register when the link partner wishes to transmit Next Page information. • 1 = Link partner has Next Page ability • 0 = Link partner does not have Next Page ability	0	R/O
14	Acknowledge	The BCM57XXreturns a 1 on bit 14 of the Link Partner Ability Register when the link partner has acknowledged reception of the link code word, otherwise, it returns a 0. • 1 = Link partner has received link code word • 0 = Link partner has not received link code word	0	R/O
13	Remote Fault	The BCM57XX returns a 1 on bit 13 of the Link Partner Ability Register when the link partner has advertised detection of a remote fault, otherwise, it returns a 0. • 1 = Link partner has detected remote fault • 0 = Link partner has not detected remote fault	0	R/O
12	Reserved Technologies	Bit 12 of the Link Partner Ability Register is reserved for future versions of the auto-negotiation standard and should be ignored when read. Write as 0, ignore on read.	0	R/O

Table 614: Auto-Negotiation Link Partner Ability Register (PHY_Addr = 0x1, Reg_Addr = 05h) (Cont.)

Bit	Field	Description	Init	Access
11	Asymmetric Pause	The BCM57XX returns a 1 on bit 11 of the Link Partner Ability Register when the link partner has advertised asymmetric pause, otherwise, it returns a 0. • 1 = Link partner desires asymmetric pause • 0 = Link partner does not desire asymmetric pause	0	R/O
10	Pause Capable	The BCM57XX returns a 1 on bit 10 of the Link Partner Ability Register when the link partner has advertised pause capability, otherwise, it returns a 0. • 1 = Link partner is capable of pause operation • 0 = Link partner not capable of pause operation	0	R/O
9	100BASE-T4 Capability	The BCM57XX returns a 1 on bit 9 of the Link Partner Ability Register when the link partner has advertised 100BASE-T4 capability, otherwise, it returns a 0. • 1 = Link partner is 100BASE-T4 capable • 0 = Link partner is not 100BASE-T4 capable	0	R/O
8	100BASE-TX Full-Duplex Capability	The BCM57XX returns a 1 on bit 8 of the Link Partner Ability Register when the link partner has advertised 100BASE-TX full-duplex capability, otherwise, it returns a 0. • 1 = Link partner is 100BASE-TX full-duplex capable • 0 = Link partner is not 100BASE-TX full-duplex capable	0	R/O
7	100BASE-TX Half-Duplex Capability	The BCM57XX returns a 1 on bit 7 of the Link Partner Ability Register when the link partner has advertised 100BASE-TX half-duplex capability, otherwise, it returns a 0. • 1 = Link partner is 100BASE-TX half-duplex capable • 0 = Link partner not 100BASE-TX half-duplex capable	0	R/O
6	10BASE-T Full-Duplex Capability	The BCM57XX returns a 1 on bit 6 of the Link Partner Ability Register when the link partner has advertised 10BASE-T full-duplex capability, otherwise, it returns a 0. • 1 = Link partner is 10BASE-T full-duplex capable • 0 = Link partner is not 10BASE-T full-duplex capable	0	R/O
5	10BASE-T Half-Duplex Capability	The BCM57XX returns a 1 on bit 5 of the Link Partner Ability Register when the link partner has advertised 10BASE-T half-duplex capability, otherwise, it returns a 0. • 1 = Link partner is 10BASE-T half-duplex capable • 0 = Link partner is not 10BASE-T half-duplex capable	0	R/O
4:0	Protocol Selector Field	Bits 4:0 of the Link Partner Ability Register return the value of the link partner's advertised protocol selector field. Link partner protocol selector field.	00000	R/O

Auto-Negotiation Expansion Register (PHY_Addr = 0x1, Reg_Addr = 06h)

Table 615: Auto-Negotiation Expansion Register (PHY_Addr = 0x1, Reg_Addr = 06h)

Bit	Field	Description	Init	Access
15:5	Reserved	Ignore when read	-	R/O
4	Parallel Detection Fault	Bit 4 of the Auto-negotiation Expansion Register returns a 1 when a parallel detection fault has occurred in the auto-negotiation state machine. When a parallel detection fault occurs, this bit is latched at 1 and remains so until the register read. This bit returns a 0 when a parallel detection fault has not occurred since the last time it was read. • 1 = Parallel link fault detected. • 0 = Parallel link fault not detected.	0	R/O LH
3	Link Partner Next Page Ability	The BCM57XX returns a 1 on bit 3 of the Auto-negotiation Expansion Register 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 Auto-negotiation Link Partner Ability Register (see "Auto-Negotiation Link Partner Ability Register (PHY_Addr = 0x1, Reg_Addr = 05h)" on page 610). • 1 = Link partner has Next Page capability. • 0 = Link partner does not have Next Page capability.	0	R/O
2	Next Page Capability	The BCM57XX returns a 1 on bit 1 of the Auto-negotiation Expansion Register 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. • 1 = BCM57XX is Next Page capable. • 0 = BCM57XX is not Next Page capable.	1	R/O LH
1	Page Received	The BCM57XX returns a 1 on bit 1 of the Auto-negotiation Expansion Register 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. • 1 = New page has been received from link partner. • 0 = New page has not been received.	0	R/O LH
0	Link Partner Auto-negotiation Ability	The BCM57XX returns a 1 on bit 0 of the Auto-negotiation Expansion Register 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. • 1 = Link partner has auto-negotiation capability. • 0 = Link partner does not have auto-negotiation.	0	R/O

Next Page Transmit Register (PHY_Addr = 0x1, Reg_Addr = 07h)

Table 616: Next Page Transmit Register (PHY_Addr = 0x1, Reg_Addr = 07h)

Bit	Field	Description	Init	Access
15	Next Page	Bit 15 of the Next Page Transmit Register must be set to 1 to indicate that more Next Pages are to be sent. This bit must be set to 0 to indicate that this is the last Next Page to be transmitted. When this bit is read, it returns the last value written. • 1 = Additional Next Pages follow. • 0 = Sending last Next Page.	0	R/W
14	Reserved	Write as 0, ignore on read.	0	R/O
13	Message Page	Bit 13 of the Next Page Transmit Register must be set to 1 to indicate that a formatted message page is being sent. This bit must be set to 0 to indicate that an unformatted page is being sent. When this bit is read, it returns the last value written. • 1 = Formatted page. • 0 = Unformatted page.	1	R/W
12	Acknowledge2	When this bit is set to 1, the BCM57XX indicates that it can comply with the Next Page request. When this bit is set to 0, the BCM57XX indicates that it cannot comply with the Next Page request. When this bit is read, it returns the last value written. • 1 = Complies with message. • 0 = Cannot comply with message.	0	R/W
11	Toggle	This bit toggles between different Next Page exchanges to insure a functional synchronization to the Link Partner. Toggles between exchanges of different Next Pages.	0	R/O
10:0	Message or Unformatted Code Field	These 11 bits make up the message code defined IEEE 802.3 section 28, Annex C when sending formatted pages. When sending unformatted Next Pages, these 11 bits contain an arbitrary data value.	0...01	R/W

Link Partner Received Next Page Register (PHY_Addr = 0x1, Reg_Addr = 08h)

Table 617: Link Partner Received Next Page Register (PHY_Addr = 0x1, Reg_Addr = 08h)

Bit	*Field*	*Description*	*Init*	*Access*
15	Next Page	Bit 15 of the Link Partner Received Next Page Register returns a 1 when the link partner has indicated that more Next Pages are to be sent. This bit returns a 0 when the link partner has indicated that this is the last Next Page to be transmitted. • 1 = Additional Next Pages follow. • 0 = Sending last Next Page.	0	R/O
14	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. • 1 = Acknowledge. • 0 = No acknowledge.	0	R/O
13	Message Page	Bit 13 of the Link Partner Received Next Page Register 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. • 1 = Formatted page. • 0 = Unformatted page.	0	R/O
12	Acknowledge2	Bit 12 of the Link Partner Received Next Page Register returns a 1 when the link partner indicates that it can comply with the Next Page request. This bit returns a 0 when the link partner indicates that it cannot comply with the Next Page request. • 1 = Complies with message. • 0 = Cannot comply with message.	0	R/O
11	Toggle	The link partner toggles this bit between different Next Page exchanges to insure a functional synchronization to the BCM57XX. Toggles between exchanges of different Next Pages.	0	R/O
10:0	Message Code Field	These 11 bits make up the message code defined IEEE 802.3, Section 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.	0...0	R/O

1000BASE-T Control Register (PHY_Addr = 0x1, Reg_Addr = 09h)

Table 618: 1000BASE-T Control Register (PHY_Addr = 0x1, Reg_Addr = 09h)

Bit	Field	Description	Init	Access
15:13	Test Mode	The BCM57XX 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. • 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.	000	R/W
12	Master/Slave Configuration Enable	When bit 12 of the 1000BASE-T Control Register is written to 1, the BCM57XX master/slave mode is configured using the manual master/slave configuration value. When the bit is written to 0, 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. • 1 = Enable Master/Slave manual configuration value. • 0 = Automatic Master/Slave configuration.	0	R/W
11	Master/Slave Configuration Value	When bit 12 of the 1000BASE-T Control Register is written to 1, bit 11 of the 1000BASE-T Control Register determines the BCM57XX master/slave mode of operation. When bit 11 is set to 1, the BCM57XX is configured as master. When bit 11 is set to 0, the BCM57XX is configured as slave. When read, this bit returns the last value written. • 1 = Configure PHY as Master. • 0 = Configure PHY as Slave.	0	R/W
10	Repeater/DTE	When bit 10 of the 1000BASE-T Control Register is written to 1, the BCM57XX advertises that it is a repeater or switch device port. When the bit is written to 0, the BCM57XX advertises that it is a DTE port. The advertised value is used in the automatic master/slave configuration resolution. The link partner that advertises repeater mode is configured to master if the opposing link partner advertises DTE, otherwise, this bit has no effect. This bit returns a 1 when advertising repeater/switch mode, otherwise, it returns a 0. By default, the device advertises that it is a DTE. • 1 = Repeater/switch device port. • 0 = DTE device.	0	R/W
9	Advertise 1000BASE-T Full-Duplex Capability	When bit 9 of the 1000BASE-T Control Register is written to 1, the BCM57XX advertises 1000BASE-T full-duplex capability. When bit 9 is written to 0, the BCM57XX advertises no 1000BASE-T full-duplex capability. This bit returns a 1 when advertising 1000BASE-T full-duplex capability, otherwise, it returns a 0. The default value of this bit is 1. • 1 = Advertise 1000BASE-T full-duplex capability. • 0 = Advertise no 1000BASE-T full-duplex capability.	1	R/W

Table 618: 1000BASE-T Control Register (PHY_Addr = 0x1, Reg_Addr = 09h) (Cont.)

Bit	Field	Description	Init	Access
8	Advertise 1000BASE-T Half-Duplex Capability	When bit 8 of the 1000BASE-T Control Register is written to 1, the BCM57XX advertises 1000BASE-T half-duplex capability. When bit 8 is written to 0, the BCM57XX advertises no 1000BASE-T half-duplex capability. This bit returns a 1 when advertising 1000BASE-T half-duplex capability, otherwise, it returns a 0. • 1 = Advertise 1000BASE-T half-duplex capability. • 0 = Advertise no 1000BASE-T half-duplex capability.	1	R/W
7:0	Reserved	Write as 0, ignore on read.	0....0	R/O

1000BASE-T Status Register (PHY_Addr = 0x1, Reg_Addr = 0Ah)

The values contained in bits 14,11, and 10 of the 1000BASE-T Status Register are only guaranteed to be valid once auto-negotiation has successfully completed, as indicated by bit 5 of the MII status register.

Table 619: 1000BASE-T Status Register (PHY_Addr = 0x1, Reg_Addr = 0Ah)

Bit	Field	Description	Init	Access
15	Master/Slave Configuration Fault	The BCM57XX returns a 1 on bit 15 of the 1000BASE-T Status Register when a master/slave configuration fault has occurred during auto-negotiation. When a configuration fault occurs, the bit is latched at 1 and remain 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. • 1 = Master/Slave configuration fault detected. • 0 = No Master/Slave configuration fault detected.	0	R/O LH
14	Master/Slave Configuration Resolution	When the BCM57XX has been configured as master, it returns a 1 on bit 14 of the 1000BASE-T Status Register. When the BCM57XX has been configured as slave, it returns a 0. • 1 = Local transmitter is Master. • 0 = Local transmitter is Slave.	0	R/O
13	Local Receiver Status	The BCM57XX returns a 1 on bit 13 of the 1000BASE-T Status Register when the local receiver status is OK, otherwise, it returns a 0. • 1 = Local receiver OK. • 0 = Local receiver not OK.	0	R/O
12	Remote Receiver Status	The BCM57XX returns a 1 on bit 12 of the 1000BASE-T Status Register when the remote receiver status is OK, otherwise, it returns a 0. • 1 = Remote receiver OK. • 0 = Remote receiver not OK.	0	R/O
11	Link Partner 1000BASE-T Full-Duplex Capability	The BCM57XX returns a 1 on bit 11 of the 1000BASE-T Status Register when the link partner has advertised 1000BASE-T full-duplex capability, otherwise, it returns a 0. • 1 = Link partner is 1000BASE-T full-duplex capable. • 0 = Link partner not 1000BASE-T full-duplex capable.	0	R/O

Table 619: 1000BASE-T Status Register (PHY_Addr = 0x1, Reg_Addr = 0Ah) (Cont.)

Bit	Field	Description	Init	Access
10	Link Partner 1000BASE-T Half-Duplex Capability	The BCM57XX returns a 1 on bit 10 of the 1000BASE-T Status Register when the link partner has advertised 1000BASE-T half-duplex capability, otherwise, it returns a 0. • 1 = Link partner is 1000BASE-T half-duplex capable. • 0 = Link partner not 1000BASE-T half-duplex capable.	0	R/O
9:8	Reserved	Write as 0, ignore on read.	0	R/O
7:0	Idle Error Count	The BCM57XX counts the number of idle errors received while the local receiver status is OK. Bits 7 through 0 of the 1000BASE-T Status Register returns 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. The Number of idle errors since last read.	0	R/O

IEEE Extended Status Register (PHY_Addr = 0x1, Reg_Addr = 0Fh)

Table 620: IEEE Extended Status Register (PHY_Addr = 0x1, Reg_Addr = 0Fh)

Bit	Field	Description	Init	Access
15	1000BASE-X Full-Duplex Capability	The BCM57XX is not capable of 1000BASE-X full-duplex operation, and returns a 0 when bit 15 of the IEEE extended Status Register is read. • 1 = 1000BASE-X full-duplex capable. • 0 = Not 1000BASE-X full-duplex capable.	0	R/O L
14	1000BASE-X Half-Duplex Capability	The BCM57XX is not capable of 1000BASE-X half-duplex operation, and returns a 0 when bit 14 of the IEEE extended Status Register is read. • 1 = 1000BASE-X half-duplex capable. • 0 = Not 1000BASE-X half-duplex capable.	0	R/O L
13	1000BASE-T Full-Duplex Capability	The BCM57XX is capable of 1000BASE-T full-duplex operation, and returns a 1 when bit 13 of the IEEE extended Status Register is read. • 1 = 1000BASE-T full-duplex capable. • 0 = Not 1000BASE-T full-duplex capable.	1	R/O H
12	1000BASE-T Half-Duplex Capability	The BCM57XX is capable of 1000BASE-T half-duplex operation, and returns a 1 when bit 12 of the IEEE extended Status Register is read. • 1 = 1000BASE-T half-duplex capable. • 0 = Not 1000BASE-T half-duplex capable.	1	R/O H
11:0	Reserved	Write as 0, ignore on read.	0	R/O

00H-0FH 1000BASE-X REGISTER MAP DETAILED DESCRIPTION

00h: 1000-X MII Control Register

Table 621: 00h: 1000-X MII Control Register

Bit	Name	R/W	Description	Default
15	RESET	R/W SC	• 1 = PHY Reset • 0 = Normal operation	0
14	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	
11	POWER DOWN	R/W	• 1 = Low-power mode • 0 = Normal operation	
10	ISOLATE	R/W	• 1 = Isolate PHY from MII • 0 = Normal operation	
9	RESTART AUTO-NEGOTIATION	R/W SC	• 1 = Restart auto-negotiation process • 0 = Normal operation	0
8	DUPLEX MODE	R/W	• 1 = Full-duplex • 0 = Half-duplex	
7	COLLISION TEST	R/W	• 1 = Collision test mode enabled • 0 = Collision test mode disabled	0
6	RESERVED	RO	Write as 1, ignore on read	1
5:0	RESERVED	RO	Write as 0, ignore on read	000000

Note: This register is applicable to BCM5714S and BCM5715S devices only.

01h: 1000-X MII Status Register

Table 622: 01h: 1000-X MII Status Register

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 register 0Fh • 0 = No extended status info in register 0Fh	1
7	RESERVED	RO	Ignore on read	0
6	MF PREAMBLE SUPPRESSION	RO H	• 1 = PHY accepts management frames with preamble suppressed • 0 = PHY does not accept management frames with preamble suppressed	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 pass • 0 = Link fail	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 supported • 0 = Basic register set capabilities only	1

Note: This register is applicable to BCM5714S and BCM5715S devices only.

04h: 1000-X Auto-Negotiation Advertisement Register

Table 623: 04h: 1000-X Auto-Negotiation Advertisement Register

Bit	Name	R/W	Description	Default
15	NEXT PAGE	RO	Write as 0, ignore on read	0
14	RESERVED	RO	Write as 0, ignore on read	0
13:12	REMOTE FAULT	R/W	• 00 = No remote fault • 01 = Link failure • 10 = Offline • 11 = Auto-negotiation error	00
11:9	RESERVED	R/W	write as 0, ignore on read	000
8:7	PAUSE	R/W	• 00 = No pause • 01 = Symmetric pause • 10 = Asymmetric pause toward link partner • 11 = Both symmetric pause and asymmetric pause toward local device	
6	HALF DUPLEX	R/W	• 1 = Advertise half-duplex • 0 = Do not advertise half-duplex	
5	FULL DUPLEX	R/W	• 1 = Advertise full-duplex • 0 = Do not advertise full-duplex	
4:0	RESERVED	R/W	write as 0, ignore on read	00000

Note: This register is applicable to BCM5714S and BCM5715S devices only.

05h: 1000-X Auto-Negotiation Link Partner Ability Register (Base Page)

Table 624: 05h: 1000-X Auto-Negotiation Link Partner Ability Register (Base Page)

Bit	Name	R/W	Description	Default
15	NEXT PAGE	RO	• 1 = Link partner is next page able • 0 = Link partner is not next page able	0
14	ACKNOWLEDGE	RO	• 1 = Link partner has received link code word • 0 = Link partner has not received link code word	0
13:12	REMOTE FAULT	RO	• 00 = No remote fault • 01 = Link failure • 10 = Offline • 11 = Auto-negotiation error	00
11:9	RESERVED	RO	Ignore on read	000
8:7	PAUSE	RO	• 00 = No pause • 01 = Symmetric pause • 10 = Asymmetric pause toward link partner • 11 = Both symmetric pause and asymmetric pause toward local device	00
6	HALF DUPLEX CAPABLE	RO	• 1 = Link partner is half-duplex capable • 0 = Link partner is not half-duplex capable	0
5	FULL DUPLEX CAPABLE	RO	• 1 = Link partner is full-duplex capable • 0 = Link partner is not full-duplex capable	0
4:0	RESERVED	RO	ignore on read	00000

Note: This register is applicable to BCM5714S and BCM5715S devices only.

06h: 1000-X Auto-Negotiation Expansion Register

Table 625: 06h: 1000-X Auto-Negotiation Expansion Register

Bit	Name	R/W	Description	Default
15:3	RESERVED	RO	Ignore on read	000h
2	NEXT PAGE ABILITY	RO L	• 1 = Local device is next page able • 0 = Local device is not next page able	0
1	PAGE RECEIVED	RO LH	• 1 = New link code word has been received • 0 = New link code word has not been received	0
0	RESERVED	RO	Ignore on read	0

Note: This register is applicable to BCM5714S and BCM5715S devices only.

07h: 1000-X Auto-Negotiation Next Page Transmit Register

Table 626: 07h: 1000-X Auto-Negotiation Next Page Transmit Register

Bit	Name	R/W	Description	Default
15:0	RESERVED	RO	Write as 0, ignore on read	0000h

Note: This register is applicable to BCM5714S and BCM5715S devices only.

08h: 1000-X Auto-Negotiation Link Partner Ability Register (Next Page)

Table 627: 08h: 1000-X Auto-Negotiation Link Partner Ability Register (Next Page)

Bit	Name	R/W	Description	Default
15:0	RESERVED	RO	Ignore on read	0000h

Note: This register is applicable to BCM5714S and BCM5715S devices only.

09h: 1000-X Reserved Register

Table 628: 09h: 1000-X Reserved Register

Bit	Name	R/W	Description	Default
15:0	RESERVED	RO	Write as 0, ignore on read	0000h

Note: This register is applicable to BCM5714S and BCM5715S devices only.

0Ah: 1000-X Reserved Register

Table 629: 0Ah: 1000-X Reserved Register

Bit	Name	R/W	Description	Default
15:0	RESERVED	RO	Ignore on read	0000h

Note: This register is applicable to BCM5714S and BCM5715S devices only.

0Fh: 1000-X Extended Status Register

Table 630: 0Fh: 1000-X Extended Status Register

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
11:0	RESERVED	RO	Ignore on read	000h

Note: This register is applicable to BCM5714S and BCM5715S devices only.

PHY Extended Control Register (PHY_Addr = 0x1, Reg_Addr = 10h)

Table 631: PHY Extended Control Register (PHY_Addr = 0x1, Reg_Addr = 10h)

Bit	Field	Description	Init	Access
15	MAC/PHY Interface Mode	The MAC/PHY interface is GMII. • 1 = TBI (10-bit Interface). • 0 = GMII.	0	R/W
14	Disable Automatic MDI Crossover	The automatic MDI crossover function can be disabled by writing a 1 to bit 14 of the PHY Extended Control Register. When the bit is written to 0, the BCM5701 performs the automatic MDI crossover function (see "Automatic MDI Crossover" on page 68). • 1 = Automatic MDI crossover disabled. • 0 = Automatic MDI crossover enabled.	0	R/W
13	Transmit Disable	The transmitter can be disabled by writing a 1 to bit 13 of the PHY Extended Control Register. The transmitter outputs (TRD±{0...3}) are forced into a high impedance state. • 1 = Transmitter outputs disabled. • 0 = Normal operation.	0	R/W
12	Interrupt Disable	• 1 = Interrupt status output disabled. • 0 = Interrupt status output enabled.	0	R/W
11	Force Interrupt	• 1 = Force interrupt status to active. • 0 = Normal operation.	0	R/W
10	Bypass 4B/5B Encoder/Decoder	The 100BASE-TX 4B/5B encoder/decoder can be bypassed by writing a 1 to bit 10: • 1 = Transmit and receive 5B codes over MII pins. • 0 = Normal MII.	0	R/W

Table 631: PHY Extended Control Register (PHY_Addr = 0x1, Reg_Addr = 10h) (Cont.)

Bit	Field	Description	Init	Access
9	Bypass Scrambler/ Descrambler	The 100BASE-TX stream cipher function can be disabled by writing a 1 to bit 9 of the PHY Extended Control Register. The stream cipher function can be re-enabled by writing a 0 to this bit. • 1 = Scrambler and descrambler disabled. • 0 = Scrambler and descrambler enabled.	0	R/W
8	Bypass MLT3 Encoder/Decoder	The 100BASE-TX MLT3 encoder and decoder can be bypassed by writing a 1 to bit 8 of the PHY Extended Control Register. NRZ data is transmitted and received on the cable. The MLT3 encoder can be re-enabled by writing a 0 to this bit. • 1 = Bypass NRZI/MLT3 encoder and decoder. • 0 = Normal operation.	0	R/W
7	Bypass Receive Symbol Alignment	100BASE-TX receive symbols alignment can be bypassed by writing a 1 to bit 7. • 1 = 5B receive symbols not aligned. • 0 = Receive symbols aligned to 5B boundaries.	0	R/W
6	Reset Scrambler	When bit 6 of the PHY Extended Control Register is written to 1, the BCM57XX resets the scrambler to an all 1s state. This bit is self-clearing, and always returns 0 when read. • 1 = Reset scrambler to all 1s state. • 0 = Normal scrambler operation.	0	R/W SC
5	Enable LED Traffic Mode	When bit 5 of the PHY Extended Control Register is written to 1, the BCM57XX enables the LED traffic mode. When bit 5 is written to 0, the BCM57XX disables the LED traffic mode. In this mode, the traffic LED will blink faster with a higher rate of traffic, and will stay on during heavy traffic. • 1 = LED traffic mode enabled. • 0 = LED traffic mode disabled.	0	R/W
4	Force LEDs ON	When bit 4 of the PHY Extended Control Register is written to 1, the BCM57XX forces all LEDs into the ON state. When bit 4 is written to 0, the BCM57XX resets all LEDs to normal operation. • 1 = Force all LEDs into ON state. • 0 = Normal LED operation.	0	R/W
3	Force LEDs OFF	When bit 3 of the PHY Extended Control Register is written to 1, the BCM57XX forces all LEDs into the OFF state. When bit 3 is written to 0, the BCM57XX resets all LEDs to normal operation. • 1 = Force all LEDs into OFF state. • 0 = Normal LED operation.	0	R/W
2	Reserved (BCM5705, BCM5721, and BCM5751 only)	-		
	Extend Transmit IPG Mode (other devices)	When bit 2 of the PHY Extended Control Register is written to 1, the BCM57XX extends the transmit IPG to at least four nibbles in 100BASE-TX mode. When bit 2 is written to 0, the BCM57XX does not extend short transmit IPGs. • 1 = Extend transmit IPGs at least four nibbles in 100BASE-TX mode. • 0 = Do not extend short transmit IPGs.	0	R/W

Table 631: PHY Extended Control Register (PHY_Addr = 0x1, Reg_Addr = 10h) (Cont.)

Bit	Field	Description	Init	Access
1	Three Link LED Mode	For the BCM5700 MAC, the following description applies: When bit 1 of the PHY Extended Control Register is written to 1, the LINK10, LINK100 and LINK1000 pins each indicate link status for the corresponding speed of operation (LINKLED indicates 10 Mbps link mode). When this bit is written to 0, the LINKLED pin indicates link status, the LNK100 pin indicates 100 Mbps speed selection, and the LNK1000 pin indicates 1000 Mbps speed selection. • 1 = Three link LED mode enabled. • 0 = Link/speed LED mode enabled.	1	R/W
		For the BCM5701 device, the following description applies: When bit 1 of the PHY Extended Control Register is written to 1, the LINKLED, SPD100LED and SPD1000LED pins each indicate link status for the corresponding speed of operation (LINKLED indicates 10 Mbps link mode). When this bit is written to 0, the LINKLED pin indicates link status, the SPD100LED pin indicates 100 Mbps speed selection, and the SPD1000LED pin indicates 1000 Mbps speed selection. • 1 = Three link LED mode enabled. • 0 = Link/speed LED mode enabled. ***Note:*** In the BCM5702 MAC Transceiver and later, this bit is reserved. Do not write to this bit because doing so will cause the MAC to malfunction.	1	R/W
0	GMII FIFO Elasticity	When bit 0 of the PHY Extended Control Register is written to 1, the BCM57XX sets the GMII Fifo Elasticity to high latency. In this mode the BCM57XX can transmit packets up to 9 KB in length. When this bit is written to 0, the GMII Fifo Elasticity is set to low latency. In this mode the BCM57XX can transmit packets up to 4.5-KB in length. Setting this bit to 1 adds 16 ns to the 1000BASE-T transmit latency. • 1 = High latency. • 0 = Low latency.	0	R/W

PHY Extended Status Register (PHY_Addr = 0x1, Reg_Addr = 11h)

Table 632: PHY Extended Status Register (PHY_Addr = 0x1, Reg_Addr = 11h)

Bit	Field	Description	Init	Access
15	Auto-negotiation Base Page Selector Field Mismatch (BCM5705, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715 only)	• 1 = Link Partner Base Page Selector field mismatched Advertised Selector field since last read. • 0 = No mismatch detected since last read.	0	RO, LH
	S3MII FIFO Error (other devices)	The BCM57XX returns a 1 on bit 15 of the PHY Extended Status Register whenever S3MII FIFO overrun occurs. This bit set to 0 after reading this register. • 1 = S3MII FIFO overflow occurred since last read. • 0 = No FIFO error detected since last read.	0	R/O
14	Reserved (BCM5705, BCM5721, and BCM5751 only)			
	Wire speed Down Grade (other devices)	When wire speed is set to a 1, if auto-negotiation is enabled and the BCM57XX is unable to establish link in five tries, then the BCM57XX automatically downgrades the speed it advertises and tries establishing a link. If Wire speed is set to a 0, no speed downgrading occurs. • 1 = Auto-negotiation Adv. speed down graded. • 0 = No Adv. speed down grading.	0	R/O
13	MDI Crossover State	The BCM57XX returns a 1 on bit 13 of the PHY Extended Status Register when the BCM57XX is automatically switching the transmit and receive pairs to communicate with a remote device. This bit returns a 0 when the BCM57XX is in normal MDI mode. • 1 = Performing MDI crossover. • 0 = Normal MDI mode.	0	R/O
12	Interrupt Status	The BCM57XX returns a 1 on bit 12 of the PHY Extended Status Register when any unmasked interrupt is currently active, otherwise, it returns a 0. • 1 = Unmasked interrupt currently active. • 0 = Interrupts clear.	0	R/O
11	Remote Receiver Status	The BCM57XX returns a 1 on bit 11 of the PHY Extended Status Register when the remote receiver status is OK. When the BCM57XX 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. • 1 = Remote receiver OK. • 0 = Remote receiver not OK since last read.	0	R/O LL

Table 632: PHY Extended Status Register (PHY_Addr = 0x1, Reg_Addr = 11h) (Cont.)

Bit	Field	Description	Init	Access
10	Local Receiver Status	The BCM57XX returns a 1 on bit 10 of the PHY Extended Status Register when the local receiver status is OK. When the BCM57XX 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. • 1 = Local receiver OK. • 0 = Local receiver not OK since last read.	0	R/O LL
9	Locked	The BCM57XX returns a 1 on bit 9 of the PHY Extended Status Register when the descrambler is locked to the incoming data stream, otherwise, it returns a 0. • 1 = Descrambler locked. • 0 = Descrambler unlocked.	0	R/O
8	Link Status	The BCM57XX returns a 1 on bit 8 of the PHY Extended Status Register when the link status is good, otherwise, it returns a 0. • 1 = Link pass. • 0 = Link fail.	0	R/O
7	CRC Error Detected	The BCM57XX returns a 1 on bit 7 of the PHY Extended Status Register while its link partner is signaling a remote fault condition, otherwise, it returns a 0. • 1 = CRC error detected. • 0 = No CRC error since last read.	0	R/O
6	Carrier Extension Error Detected	The BCM57XX returns a 1 on bit 6 of the PHY Extended Status Register if a carrier extension error has been detected since the last time this register was read, otherwise, it returns a 0. • 1 = Carrier extension error detected since last read. • 0 = No carrier extension error since last read.	0	R/O LH
5	Bad SSD Detected (False Carrier)	The BCM57XX returns a 1 on bit 5 of the PHY Extended Status Register if a bad start of stream error has been detected since the last time this register was read, otherwise, it returns a 0. • 1 = Bad SSD error detected since last read. • 0 = No bad SSD error since last read.	0	R/O LH
4	Bad ESD Detected (Premature End)	The BCM57XX returns a 1 on bit 4 of the PHY Extended Status Register if a bad end of stream error has been detected since the last time this register was read, otherwise, it returns a 0. • 1 = Bad ESD error detected since last read. • 0 = No bad ESD error since last read.	0	R/O LH
3	Receive Error Detected	The BCM57XX returns a 1 on bit 3 of the PHY Extended Status Register if a packet was received with an invalid code since the last time this register was read, otherwise, it returns a 0. • 1 = Receive error detected since last read. • 0 = No receive error since last read.	0	R/O LH

Table 632: PHY Extended Status Register (PHY_Addr = 0x1, Reg_Addr = 11h) (Cont.)

Bit	*Field*	*Description*	*Init*	*Access*
2	Transmit Error Detected	The BCM57XX returns a 1 on bit 2 of the PHY Extended Status Register if a packet was received with a transmit error code since the last time this register was read, otherwise, it returns a 0. • 1 = Transmit error code received since last read. • 0 = No transmit error code received since last read.	0	R/O LH
1	Lock Error Detected	The BCM57XX returns a 1 on bit 1 of the PHY Extended Status Register if the descrambler has lost lock since the last time this register was read, otherwise, it returns a 0. • 1 = Lock error detected since last read. • 0 = No lock error since last read.	0	R/O LH
0	MLT3 Code Error Detected	The BCM57XX returns a 1 on bit 0 of the PHY Extended Status Register 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. • 1 = MLT3 code error detected since last read. • 0 = No MLT3 code error since last read.	0	R/O LH

RECEIVE ERROR COUNTER (PHY_ADDR = 0X1, REG_ADDR = 12H)

This counter increments each time the BCM57XX family receivers a non-collision packet containing at least one receive error (freezes at the maximum value FFFFh). The counter automatically clears when read.

Table 633: Receive Error Counter (PHY_Addr = 0x1, Reg_Addr = 12h)

Bit	*Field*	*Description*	*Init*	*Access*
15:0	Receive Error Counter	Number of non-collision packets with receive errors since last read.	0000h	R/W

FALSE CARRIER SENSE COUNTER (PHY_ADDR = 0X1, REG_ADDR = 13H)

S3MII Error Counter increments each time the BCM57XX family detects a S3MII overrun/underrun event. False Carrier Sense Counter increments each time the BCM57XX family detects a false carrier on the receive input. These counters freeze at the maximum value FFh. The counters automatically clear when read.

Table 634: False Carrier Sense Counter (PHY_Addr = 0x1, Reg_Addr = 13h)

Bit	*Field*	*Description*	*Init*	*Access*
15:8	Reserved (BCM5705, BCM5721, BCM5751, and BCM5714 only)	-		
	SMII FIFO Error Counter (other devices)	Number of SMII overrun/underrun events since last read. Freezes at 0FFh.	00h	R/W
7:0	False Carrier Sense Counter	Number of false carrier sense events since last read.	00h	R/W

RECEIVER NOT_OK COUNTERS (PHY_ADDR = 0X1, REG_ADDR = 14H)

Normal Operation (CRC Count Visibility = 0)

These counters increment each time the local or remote receiver enters the NOT_OK state (freezes at the maximum value FFh) when the CRC Error Count Visibility bit of PHY Test Register 1 (see "PHY Test Register 1 (PHY_Addr = 0X1, REG_Addr = 1EH)" on page 700) is clear. The counters automatically clear when read.

Table 635: Receiver NOT_OK Counters (PHY_Addr = 0x1, Reg_Addr = 14h, Normal Operation)

Bit	Field	Description	Init	Access
15:8	Local Receiver NOT_OK Counter	Number of times local receiver was not OK since last read (when PHY Test Register 1. CRC_Error_Count_Visibility bit (see "PHY Test Register 1 (PHY_Addr = 0X1, REG_Addr = 1EH)" on page 700) is clear).	00h	R/W
7:0	Remote Receiver NOT_OK Counter	Number of times BCM57XX detected that the remote receiver was not OK since last read (when PHY Test Register 1. CRC_Error_Count_Visibility bit (see "PHY Test Register 1 (PHY_Addr = 0X1, REG_Addr = 1EH)" on page 700) is clear).	00h	R/W

CRC Error Count Operation (CRC Count Visibility = 1)

The CRC error counter is merged into a 16-bit counter and increments each time the BCM5701 MAC Transceiver detects a CRC error when the CRC Error Count Visibility bit of PHY Test Register 1 (see "PHY Test Register 1 (PHY_Addr = 0X1, REG_Addr = 1EH)" on page 700) is set. This counter freezes at the maximum value FFFFh. The counter automatically clears when read.

Table 636: Receiver NOT_OK Counters (PHY_Addr = 0x1, Reg_Addr = 14h, CRC Error Count Operation)

Bit	Field	Description	Init	Access
15:0	CRC Error Counter	This register becomes a 16-bit CRC error counter when PHY Test Register 1.CRC_Error_Count_Visibility bit (see "PHY Test Register 1 (PHY_Addr = 0X1, REG_Addr = 1EH)" on page 700) is set.	0000h	R/W

EXPANSION REGISTERS (BCM5705, BCM5721, BCM5751, BCM5752, BCM5714, AND BCM5715 ONLY)

Expansion Register Access Data (PHY_ADDR = 01h, REG_ADDR = 15h)

When the "Expansion Register Access Register (PHY_ADDR = 0x1, Reg_Addr = 17h)" on page 630 is enabled, this register allows read/write access to the Expansion Register selected in the Expansion Register Access Register.

Expansion Register Access Register (PHY_ADDR = 0x1, Reg_Addr = 17h)

When enabled, this register serves as an index to the expansion registers. The value of the expansion register can be read/ written through register 15h (see "Expansion Register Access Data (PHY_ADDR = 01h, REG_ADDR = 15h)" on page 630). These bits should be cleared when the Expansion Register is not accessed.

Table 637: Expansion Register Access Register (PHY_ADDR = 0x1, Reg_Addr = 17h)

Bit	Field	Description	Init	Access
15:12	Reserved	-	0000	R/W
11:8	Expansion Register Select	• 1111 = Expansion Register select • XXXX = Reserved	0h	
7:0	Expansion Register Accessed	Sets the Expansion Register number accessed when reading or writing "Expansion Register Access Data (PHY_ADDR = 01h, REG_ADDR = 15h)" on page 630.		

Table 638: Expansion Register Select Values

Expansion Register	Register Name
00h	"Expansion Register 00h: Receive/Transmit Packet Counter" on page 631
01h	"Expansion Register 01h: Expansion Interrupt Status" on page 631
03h	"Expansion Register 03h: SerDes Control" on page 632
04h	"Expansion Register 04h: Multicolor LED Selector" on page 633
05h	"Expansion Register 05h: Multicolor LED Flash Rate Controls" on page 634
06h	"Expansion Register 06h: Multicolor LED Programmable Blink Controls" on page 635
10h	"Expansion Register 10h: Cable Diagnostic Controls" on page 636
11h	"Expansion Register 11h: Cable Diagnostic Results" on page 637
12h	"Expansion Register 12h: Cable Diagnostic Lengths Channels 1/2" on page 638
13h	"Expansion Register 13h: Cable Diagnostic Lengths Channels 3/4" on page 639

Expansion Register 00h: Receive/Transmit Packet Counter

The following expansion registers are enabled by writing to "Expansion Register Access Register (PHY_ADDR = 0x1, Reg_Addr = 17h)" on page 630 bits [11:0] = F00h, and read/write access is through register 15h.

Table 639: Expansion Register 00h: Receive/Transmit Packet Counter

Bit	*Field*	*Description*	*Init*	*Access*
15:0	Packet Counter (Copper Only)	Returns the packet count	0000h	R/W SC

Packet Counter (Copper Only)

When the Packet Counter Mode bit field = 1, in the "Auxiliary Control Register (PHY_Addr = 0x1, Reg_Addr = 18h, Shadow = 111, Misc Control)" on page 655, the number of receive packets is counted.

When the Packet Counter Mode bit field = 0, in the "Auxiliary Control Register (PHY_Addr = 0x1, Reg_Addr = 18h, Shadow = 111, Misc Control)" on page 655, the number of transmit packets is counted. This counter is clear on read and freezes at FFFFh.

Expansion Register 01h: Expansion Interrupt Status

The following expansion registers are enabled by writing to "Expansion Register Access Register (PHY_ADDR = 0x1, Reg_Addr = 17h)" on page 630 bits [11:0] = F01h, and read/write access is through register 15h.

Table 640: Expansion Register 01h: Expansion Interrupt Status

Bit	*Field*	*Description*	*Init*	*Access*
15:8	Reserved	Write as 00h, ignore on read.	00h	RO
7	Mode Select Change	• 1 = Mode select change detected (clears on read). • 0 = Mode select change not detected.	0	RO LH
6	SerDes Link Status Change	• 1 = SerDes link status change detected (clears on read). • 0 = Mode select change not detected.	0	RO LH
5	1000x rudi_c detected (SerDes auto-negotiation code word received) (BCM5714 and BCM5715 only)	• 1 = 1000x rudi_c detected since last read • 0 = no 1000x rudi_c detected since last read	0	RO LH
	Expansion Interrupt Status (Other Devices)	• 1 = Interrupt condition detected. • 0 = Interrupt condition not detected.	0	RO LH
4:1		• 1 = Interrupt condition detected. • 0 = Interrupt condition not detected.	0	RO LH
0	Transmit CRC Error (Copper Only)	• 1 = Transmit CRC error detected since last read. • 0 = No Transmit CRC error detected since last read.	0	RO LH

Mode Select Change

This bit indicates that mode select change is detected.

SerDes Link Status Change

This bit indicates a SerDes link status change is detected.

Expansion Interrupt Status

These bits corresponds to the Expansion Interrupt Status bits.

Transmit CRC Error (Copper Only)

Bit 0 = 1 indicates that a transmit CRC error occurred since the register was last read, otherwise, it returns a 0.

Expansion Register 03h: SerDes Control

The following expansion registers are enabled by writing to “Expansion Register Access Register (PHY_ADDR = 0x1, Reg_Addr = 17h)” on page 630 bits [11:0] = F03h, and read/write access is through register 15h.

Table 641: Expansion register 03h: SerDes Control

Bit	*Field*	*Description*	*Init*	*Access*
15	Reserved	Write as 0, ignore on read.	0	RO
14:2	Reserved	Write as 1031h, ignore on read.	1031h	R/W
1	Clock Pad Disable	• 1 = Disable SGMII clock pads SCLK± • 0 = Enable SGMII clock pads SCLK±	0	R/W
0	Reserved	Write as 0, ignore on read.	0	RO

Clock Pad Disable

This bit disables SGMII clock pads SCLK±. The default is to have the SCLK± enabled.

Expansion Register 04h: Multicolor LED Selector

The following expansion registers are enabled by writing to "Expansion Register Access Register (PHY_ADDR = 0x1, Reg_Addr = 17h)" on page 630 bits [11:0] = F04h, and read/write access is through register 15h.

Table 642: Expansion Register 04h: Multicolor LED Selector

Bit	Field	Description	Init	Access
15:10	Reserved	Write as 00h, ignore on read.	00h	R/W
9	Flash Now	1 = Initiate a multicolor LED flash. This only works when the multicolor selector is set to 0111.	0	R/W SC
8	In Phase	• 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	R/W
7:4	MULTICOLOR[2] Multicolor Selector	Selects the multicolor mode for MULTICOLOR[2]. • 0000: Encoded link/activity LED. • 0001: Encoded speed LED. • 0010: Activity flash LED. • 0011: Full-duplex LED. • 0100: Forced off. • 0101: Forced on. • 0110: Alternating LED (toggling between the modes at 50% duty cycle with a 320 ms period). • 0111: Flashing LED (toggling between the modes with an 80 ms period). • 1000: Link LED. • 1001: Activity LED. • 1010: Programmable blink LED.	0h	R/W
3:0	MULTICOLOR[1] Multicolor Selector	Selects the multicolor mode for MULTICOLOR[1]. • 0000: Encoded link/activity LED. • 0001: Encoded speed LED. • 0010: Activity flash LED. • 0011: Full-duplex LED. • 0100: Forced off. • 0101: Forced on. • 0110: Alternating LED (toggling between the modes at 50% duty cycle with a 320 ms period). • 0111: Flashing LED (toggling between the modes with an 80 ms period). • 1000: Link LED. • 1001: Activity LED. • 1010: Programmable blink LED.	0h	R/W

Flash Now

Setting this bit = 1 causes a single flash to occur on MULTICOLOR[1] when MULTICOLOR[1] Multicolor Selector is set to 0111, and a single flash to occur on LED2 when LED2 Multicolor Selector is set to 0111.

In Phase

When both MULTICOLOR[2:1] are set to the same mode, the outputs of MULTICOLOR[1] and MULTICOLOR[2] both toggle at the same time. This bit determines whether the LEDs are identical to each other or inverse from each other. When the two LED pins are attached to a special multicolored LED, the resulting LED color alternates between Off/Amber (In Phase) or Red/Green (Out of Phase).

MULTICOLOR[2] Multicolor 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 other than the operation mentioned above, such as Red/Amber, the user can put one of the Multicolor selectors to the desired toggle mode and the other Multicolor selector to Forced On.

MULTICOLOR[1] Multicolor Selector

Bits [3:0] select the Multicolor LED mode for MULTICOLOR[1]. 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 other than the operation mentioned above, such as Red/Amber, the user can put one of the Multicolor selectors to the desired toggle mode and the other Multicolor selector to Forced On.

Expansion Register 05h: Multicolor LED Flash Rate Controls

The following expansion registers are enabled by writing to "Expansion Register Access Register (PHY_ADDR = 0x1, Reg_Addr = 17h)" on page 630 bits [11:0] = F05h, and read/write access is through register 15h.

Table 643: Expansion Register 05h: Multicolor LED Flash Rate Controls

Bit	*Field*	*Description*	*Init*	*Access*
15:12	Reserved	Write as 0h, ignore on read.	0h	R/W
11:6	Alternating Rate	Determines the width and gap for Multicolor LED selector 0110 (Alternating LED mode). • 00h = 21 ms width, 21 ms gap. • 01h = 42 ms width, 42 ms gap. • 02h = 63 ms width, 63 ms gap. • ... • 07h = 168 ms width, 168 ms gap. • ... • 3Fh = 1.344s.	07h	R/W
5:0	Flash Rate	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). • 00h = 21 ms width, 21 ms gap. • 01h = 42 ms width, 42 ms gap. • 02h = 63 ms width, 63 ms gap. • ... • 3Fh = 1.344s.	01h	R/W

Alternation Rate

Setting bits [11:6] changes the width and gap of the alternating LED modes. These bits are only valid when the MULTICOLOR[1] Multicolor Selector and or the MULTICOLOR[2] Multicolor Selector bits = 0110. LED's duty cycle 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] Multicolor Selector and/or the MULTICOLOR[2] Multicolor Selector bits = 0000, 0010, or 0111. The duty cycle of the flash rate is not exactly 50%.

Expansion Register 06h: Multicolor LED Programmable Blink Controls

The following expansion registers are enabled by writing to "Expansion Register Access Register (PHY_ADDR = 0x1, Reg_Addr = 17h)" on page 630 bits [11:0] = F06h, and read/write access is through register 15h.

Table 644: Expansion Register 06h: Multicolor LED Programmable Blink Controls

Bit	*Field*	*Description*	*Init*	*Access*
15:6	Reserved	Write as 000h, ignore on read.	000h	R/W
5	Blink Update Now	• 1 = Change to the new Blink Rate now. • 0 = Wait 1s before changing Blink Rate. Controls when a change in the Blink Rate is actually displayed on the Programmable Blink LED.	0	R/W
4:0	Blink Rate	Programs the number of blinks per second that the Programmable Blink LED blinks. • 00000 = No blink. • 00001 = One blink per second. • 00010 = Two blinks per second. • 00011 = Three blinks per second. • ... • 11111 = 31 blinks per second.	00000	R/W

Blink Update Now

Setting bit 5 updates the blink rate immediately. Clearing this bit causes the blink rate to be updated after the 1s interval timer expires. This bit is only valid when the MULTICOLOR[1] Multicolor Selector and or the MULTICOLOR[2] Multicolor Selector bits = 0000, 0010, or 0111.

Blink Rate

Setting bits [4:0] determines the blink rate of the Programmable Blink LED. These bits are only valid when the MULTICOLOR[1] Multicolor Selector and or the MULTICOLOR[2] Multicolor Selector bits = 0000, 0010, or 0111.

Expansion Register 10h: Cable Diagnostic Controls

The following expansion registers are enabled by writing to “Expansion Register Access Register (PHY_ADDR = 0x1, Reg_Addr = 17h)” on page 630 bits [11:0] = F10h, and read/write access is through register 15h.

Table 645: Expansion Register 10h: Cable Diagnostic Controls

Bit	Field	Description	Init	Access
15:3	Reserved	Write as 0000h, ignore on read.	0000h	R/W
2	Natural Link	• 1 = Skips checking of open/short. • 0 = Enables checking of open/short.	0	R/W
1	Cable Diag Begin	• 1= Begins cable diagnostic algorithm. • 0 = Cable diagnostic algorithm is not started.	0	R/W SC
0	Cable Diag Mode	• 1 = Cable diagnostic mode enabled. • 0 = Cable diagnostic mode not enabled.	0	R/W

Natural Link

Setting bit 2 = 1, skips the Open and Shorts check when in Cable Diagnostic Mode.

Cable Diag Begin

Setting bit 1 =1, begins the Cable Diagnostic Algorithm when in Cable Diagnostic Mode.

Cable Diag Mode

Setting bit 0 = 1, enables Cable Diagnostic Mode. See “Expansion Register 11h: Cable Diagnostic Results” on page 637, “Expansion Register 12h: Cable Diagnostic Lengths Channels 1/2” on page 638, and “Expansion Register 13h: Cable Diagnostic Lengths Channels 3/4” on page 639, for more details.

Expansion Register 11h: Cable Diagnostic Results

The following Expansion registers are enabled by writing to "Expansion Register Access Register (PHY_ADDR = 0x1, Reg_Addr = 17h)" on page 630 bits [11:0] = F11h, and read/write access is through register 15h.

Table 646: Expansion Register 11h: Cable Diagnostic Results

Bit	Field	Description	Init	Access
15:12	Error	Per channel: An error has occurred. • Bit 15 = 1: Error has occurred on Channel 4. • Bit 15 = 0: Error has not occurred on Channel 4. • Bit 14 = 1: Error has occurred on Channel 3. • Bit 14 = 0: Error has not occurred on Channel 3. • Bit 13 = 1: Error has occurred on Channel 2. • Bit 13 = 0: Error has not occurred on Channel 2. • Bit 12 = 1: Error has occurred on Channel 1. • Bit 12 = 0: Error has not occurred on Channel 1.	0h	RO
11:8	Open/Short Found	Per channel: An open or short has been found. • Bit 11 = 1: Open/Short found on Channel 4. • Bit 11 = 0: Open/Short not found on Channel 4. • Bit 10 = 1: Open/Short found on Channel 3. • Bit 10 = 0: Open/Short not found on Channel 3. • Bit 9 = 1: Open/Short found on Channel 2. • Bit 9 = 0: Open/Short not found on Channel 2. • Bit 8 = 1: Open/Short found on Channel 1. • Bit 8 = 0: Open/Short not found on Channel 1.	0h	RO
7:4	Finished	Per channel: Open/Short and length checking have been completed successfully. • Bit 7 = 1: Open/Short/Length checking completed on Channel 4. • Bit 7 = 0: Open/Short/Length checking not completed on Channel 4. • Bit 6 = 1: Open/Short/Length checking completed on Channel 3. • Bit 6 = 0: Open/Short/Length checking not completed on Channel 3. • Bit 5 = 1: Open/Short/Length checking completed on Channel 2. • Bit 5 = 0: Open/Short/Length checking not completed on Channel 2. • Bit 4 = 1: Open/Short/Length checking completed on Channel 1. • Bit 4 = 0: Open/Short/Length checking not completed on Channel 1.	0h	RO
3:2	Reserved	Write as 00, ignore on read.	00	RO
1	Any Open/Short Found	An open or short has been found on at least one channel.	0	RO
0	All Finished	All channels have completed processing.	0	RO

Packet Counter

Bits [15:12] reflect the cable diagnostic error status of the four channels.

Open/Short Found

Bits [11:8] reflect the open and short status of the four channels.

Finished

Bits [7:4] reflect whether Open/Short and Length checking have been completed successfully.

All Finished

Bit [0] reflects whether all four channels have completed cable diagnostic processing.

Expansion Register 12h: Cable Diagnostic Lengths Channels 1/2

The Following Expansion registers are enabled by writing to "Expansion Register Access Register (PHY_ADDR = 0x1, Reg_Addr = 17h)" on page 630 bits [11:0] = F12h, and read/write access is through register 15h.

Table 647: Expansion Register 12h: Cable Diagnostic Lengths Channels1/2

Bit	Field	Description	Init	Access
15:8	Channel 2 Length	Channel 2 Open/Short Length or Cable Length (meters). • 00000000 = 0m. • 00000001 = 1m. • 00000010 = 2m. • ... • 10000000 = 128m.	00h	RO
7:0	Channel 1 Length	Channel 1 Open/Short Length or Cable Length (meters). • 00000000 = 0m. • 00000001 = 1m. • 00000010 = 2m. • ... • 10000000 = 128m.	00h	RO

Channel 2 Length

When the BCM57XX detects an open or short on the cable, bits [15:8] reflect how far away the open or short is from the BCM57XX. When no open or short is detected, bits [15:8] reflect the cable length connected to channel 2 in meters.

Channel 1 Length

When the BCM57XX detects an open or short on the cable, bits [7:0] reflect how far away the open or short is from the BCM57XX. When no open or short is detected, bits [7:0] reflect the cable length connected to channel 1 in meters.

Expansion Register 13h: Cable Diagnostic Lengths Channels 3/4

The Following Expansion registers are enabled by writing to “Expansion Register Access Register (PHY_ADDR = 0x1, Reg_Addr = 17h)” on page 630 bits [11:0] = F13h, and read/write access is through register 15h.

Table 648: Expansion Register 13h: Cable Diagnostic Lengths Channels 3/4

Bit	Field	Description	Init	Access
15:8	Channel 4 length	Channel 4 Open/Short Length or Cable Length (meters). • 00000000 = 0m. • 00000001 = 1m. • 00000010 = 2m. • ... • 10000000 = 128m.	00h	RO
7:0	Channel 3 length	Channel 3 Open/Short Length or Cable Length (meters). • 00000000 = 0m. • 00000001 = 1m. • 00000010 = 2m. • ... • 10000000 = 128m.	00h	RO

Channel 4 Length

When the BCM57XX detects an open or short on the cable, bits [15:8] reflect how far away the open or short is from the BCM57XX. When no open or short is detected, bits [15:8] reflect the cable length connected to channel 2 in meters.

Channel 3 Length

When the BCM57XX detects an open or short on the cable, bits [7:0] reflect how far away the open or short is from the BCM57XX. When no open or short is detected, bits [7:0] reflect the cable length connected to channel 2 in meters.

AUXILIARY CONTROL REGISTER (BCM5714 AND BCM5715 DEVICES ONLY)

Auxiliary Control Register (Shadow Register Selector = 000)

Table 649: 18h: Aux. Control Reg. (Shadow Reg. Selector = 000; BCM5714 and BCM5715 Only)

Bit	Field	Description	Init	Access
15	External Loopback	• 1 = external loopback enabled • 0 = normal operation	0	R/W
14	Extended Packet Length	• 1 = allow reception of extended length packets • 0 = allow normal length Ethernet packets only	0	R/W
13:12	Edgerate Control (1000t)	• 00 = 4.0ns (1000T) • 01 = 5.0ns (1000T) • 10 = 3.0ns (1000T) • 11 = 0.0ns (1000T)	00	R/W
11	Enable Sm_DSP Clock	• 1 = Clock is enabled. • 0 = Clock is gated off.	0	R/W
10	Transmit 6dB Coding	• 1 = transmit using 6dB coding • 0 = transmit using 3dB coding	1	R/W
9:8	Receive Slicing	• 00 = normal Viterbi/DFE MLSE • 01 = 4D symbol by symbol slicing for 3 dB option • 10 = 3 level 1D symbol by symbol slicing • 11 = 5 level 1D symbol by symbol slicing during • SEND IDLE/DATA, 3 level else	00	R/W
7	Disable Partial Response Filter	• 1 = transmitter partial response filter disabled • 0 = transmitter partial response filter enabled	0	R/W
6	Disable Inverse Prf	• 1 = receiver inv. partial response filter disabled (overrides Phy Control and other MII register settings if disabled) • 0 = receiver inv. partial response filter enabled	0	R/W
5:4	Edgerate Control (100TX) LSB or'ed Ed With Er Pin)	• 00 = 4.0 ns (100TX) • 01 = 5.0 ns (100TX) • 10 = 3.0 ns (100TX) • 11 = 0.0 ns (100TX)	00	R/W
3	Diagnostic Mode	• 1 = When convergence fails, hold in failed state until cleared. • 0 = Normal operation, retrain on failure	0	R/W

Table 649: 18h: Aux. Control Reg. (Shadow Reg. Selector = 000; BCM5714 and BCM5715 Only) (Cont.)

Bit	Field	Description	Init	Access
2:0	Shadow Register Selector (These bits are written on all writes to 18h regardless of the value)	• 000 = Normal operation • 001 = 10 Base-T register • 010 = Power Control register • 011 = Reserved • 100 = Misc Test register 1 • 101 = Misc Test register 2 • 110 = Reserved • 111 = Misc Control register ***Note:*** Writes to the selected shadow register are done on a single cycle (no setup required). Reads are selected by first writing to register 18h, shadow 7, bits 14:12.	000	R/W

10Base-T Register (Shadow Register Selector = 001)

Table 650: 18h: 10BASE-T Register (Shadow Register Selector = 001; BCM5714 and BCM5715 Only)

Bit	Field	Description	Init	Access
15	Manchester Code Error	• 1 = Manchester code error (10Base-T) • 0 = no Manchester code error	0	RO LH
14	Eof Error	• 1 = EOF detection error (10Base-T) • 0 = no EOF detection error	0	RO LH
13	Polarity Error	• 1 = channel polarity inverted • 0 = channel polarity correct	0	RO
12	Block RXDV Extension (LPG)	• 1 = block rxdv for 4 additional rxc cycles for LPG • 0 = normal operation	0	R/W
11	10bt TXC Invert Mode	• 1 = invert TXC output • 0 = normal operation	0	R/W
10	Class A/B Line Driver Select (ClassB_BTt)	• 1 = select class A line driver • 0 = select class B line driver	0	R/W
9:	Jabber Disable	• 1 = Jabber function disabled • 0 = Jabber function enabled	0	R/W
8	1000Base-T Signal Detect Threshold	• 1 = low signal detect threshold • 0 = high signal detect threshold	0	R/W
7	10Base-T Signal Detect Threshold	• 1 = low signal detect threshold • 0 = high signal detect threshold	0	R/W
6	10Base-T Echo Mode	• 1 = echo transmit data to receive data • 0 = normal operation	0	R/W
5	SQE Enable Mode	• 1 = enable SQE • 0 = disable SQE	0	R/W
4	10Base-T No Dribble	• 1 = correct 10BT dribble nibble • 0 = normal operation	0	R/W
3	10Base-T Serial Mode	• 1 = enable 10Base-T serial mode • 0 = normal operation	0	R/W

Table 650: 18h: 10BASE-T Register (Shadow Register Selector = 001; BCM5714 and BCM5715 Only) (Cont.)

Bit	Field	Description	Init	Access
2:0	Shadow Register Selector	Writes to the selected shadow register are done on a single cycle (no setup required). Reads are selected by first writing to register 18h, shadow 7, bits 14:12.	001	R/W

Power/MII Control Register (Shadow Register Selector = 010)

Table 651: 18h: Power/MII Control Reg. (Shadow Reg. Selector = 010; BCM5714 and BCM5715 Only)

Bit	Field	Description	Init	Access
15:13	Class A/B Mode	Controls amount of Class A vs. Class A/B TXDAC operation. • 000 = 100% class A, • 111 = most class AB.	000	R/W
12	Gigabit TXDAC Class B Mode	• 1 = Low power Class B operation • 0 = Normal Class A operation	0	R/W
11	VREG LPWR Enable	Voltage Regulator low power enable • 1 = forces vregD to lowest setting in 10T and 100TX modes	0	R/W
10:9	Digital Voltage Regulator Output Voltage (VregD)	• 00 = 1.0v • 01 = 1.1v • 10 = 1.2v • 11 = 1.3v ***Note:*** This value is overridden to “00” during 10T 100TX modes (when bit 11 is ‘1’)	10	R/W
8:7	Analog VREG Control (VegA)	• 00 = 1.0v • 01 = 1.1v • 10 = 1.2v • 11= 1.3v	10	R/W
6	Class A/B Enable	• 1 = enable Class A/B TXDAC operation • 0 = normal operation	0	R/W
5	Super Isolate	• 1 = isolate mode with no link pulses transmitted • 0 = normal operation	0	R/W
4	100TX TXDAC Class B Mode	• 1 = Low power Class B operation • 0 = Normal Class A operation	0	R/W
3	Wake On Lan	• 1 = enable wake on LAN operation • 0 = normal operation	0	R/W
2:0	Shadow Register Selector	Writes to the selected shadow register are done on a single cycle (no setup required). Reads are selected by first writing to register 18h, shadow 7, bits 14:12.	010	R/W

Miscellaneous Test Register 1 (Shadow Register Selector = 100)

Table 652: 18h: Misc. Test Register 1 (Shadow Register Selector = 100; BCM5714 and BCM5715 Only)

Bit	Field	Description	Init	Access
15	Remote Loopback Enable	• 1 = enable loopback from MDI (cable end) receive packet, through pcs and back to MDI transmit packet • 0 = disable loopback	0	R/W
14	TDX Fix Enable	1 = TDK fix (extend EOP on transmit 10BT packets)	0	R/W
13	Enable Dedicated 10Base-T DLL Bypass Clock	• 1 = 10BT dll bypass clock generated from tpin10 in dll bypass mode • 0 = 10BT dll bypass clock generated from inverted xtali input in dll bypass mode (BASET, ADC10BT, PC10BT, CRS10BT, DAC10_100 test modes use tpin10; register value ignored)	0	R/W
12	Block 10Base-T Restart Auto-Negotiation	• 1 = prevent 10BT from restarting auto-negotiation in order to break the link • 0 = normal operation	0	R/W
11	Remote Loopback Tristate	• 1 = tristate the receive MII pins (CRS, RXDV, RXD, etc.) when Remote Loopback is enabled • 0 = Remote Loopback packets appear on MII	0	R/W
10	10Base-T Wakeup	• 1 = enable 10BT dac	0	R/W
9	10Base-T Polarity Bypass	• 1 = enable polarity bypass • 0 = normal operation	0	R/W
8	10Base-T Idle Bypass	• 1 = enable idle bypass • 0 = normal operation	0	R/W
7	10Base-T Clock Reset Enable	• 1 = clock reset controlled from tpin11 • 0 = normal operation	0	R/W
6	10Base-T Bypass ADC	• 1 = bypass 10BT adc • 0 = normal operation	0	R/W
5	10Base-T Bypass CRS	• 1 = bypass 10BT crs • 0 = normal operation	0	R/W
4	Swap RXMDIX	• 1 = rx and tx operate on same pair • 0 = normal operation	0	R/W
3	10Base-T Halfout	• 1 = transmit 10BT at half amplitude • 0 = normal operation	0	R/W
2:0	Shadow Register Selector	Writes to the selected shadow register are done on a single cycle (no setup required). Reads are selected by first writing to register 18h, shadow 7, bits 14:12.	100	R/W

Miscellaneous Test Register 2 (Shadow Register Selector = 101)

Table 653: 18h: Misc. Test Register 2 (Shadow Register Selector = 101; BCM5714 and BCM5715 Only)

Bit	Field	Description	Init	Access
15:9	Reserved	write as 0, ignore on read	0000000	R/W
8	Disable Auto Encoding Correction	• 0 = Auto encoding correction enabled (overrides bits 6 & 7) • 1 = Auto encoding correction disabled	0	R/W
7	Old PCS Encoding RX	• 0 = Select IEEE compliant PCS encoding (for PCS receive) • 1 = Select old PCS encoding (for PCS receive)	0	R/W
6	Old PCS Encoding TX	• 0 = Select IEEE compliant PCS encoding (for PCS receive) • 1 = Select old PCS encoding (for PCS receive)	0	R/W
5:3	Spare	write as 0, ignore on read	000	R/W
2:0	Shadow Register Selector	Writes to the selected shadow register are done on a single cycle (no setup required). Reads are selected by first writing to register 18h, shadow 7, bits 14:12.	101	R/W

Miscellaneous Control Register (Shadow Register Selector = 111)

Table 654: 18h: Miscellaneous Control Register (Shadow Register Selector = 111)

Bit	Field	Description	Init	Access
15	Write Enable Bits	• 1 = write bits [8:3] • 0 = only write bits [14:12]	0	R/W SC
14:12	Shadow Register Read Selector	• 000 = shadow register 0 read select • 001 = shadow register 1 read select • ... • 111 = shadow register 7 read select	000	R/W
11	Packet Counter Mode	• 1 = count packets received • 0 = count packets transmitted	0	R/W
10	Bypass Wirespeed Timer	• 1 = Link fail counter will clear as soon as link is up • 0 = Link must be up for at least 2.5 seconds otherwise link fail counter will increment. ***Note:*** Can be set only if gphy port wirespd_timer_disable = 0.	0	R/W
9	Force Auto MDIX Mode	• 1 = Auto-mdix will operate when autoneg is disabled via reg 0.12 • 0 = Auto-mdix is disabled when autoneg is disabled via reg 0.12	0	R/W
8	RGMII Timing Mode	• 1 = clock delayed 90 degrees • 0 = clock and data aligned	0	R/W
7	RGMII Mode	• 1 = use reduced GMII mode • 0 = normal GMII/MII operation	0	R/W
6	RGMII RXER Mode	• 1 = mux rx_er with rx_dv for RGMII mode • 0 = mux crs with rx_dv for RGMII mode	1	R/W

Table 654: 18h: Miscellaneous Control Register (Shadow Register Selector = 111) (Cont.)

Bit	Field	Description	Init	Access
5	RGMII Out-of-band Status Disable	• 1 = send regular rx data during IPG • 0 = send out-of-band status info in RGMII mode	1	R/W
4	Wirespeed Enable	• 1 = enable wirespeed mode • 0 = normal operation • Note: Can be set only if gphy port wirespd_enable = 1.	0	R/W
3	MDIO All PHY Select	• 1 = all phy selected during mdio writes when the phy address = 00000 • 0 = normal operation	0	R/W
2:0	Shadow Register Selector (Reference Only)	Writes to the selected shadow register are done on a single cycle (no setup required). Reads are selected by first writing to register 18h, shadow 7, bits 14:12.	111	R/W

AUXILIARY CONTROL REGISTER (EXCEPT BCM5714 AND BCM5715 DEVICES)

Auxiliary Control Register (PHY_Addr = 0x1, Reg_Addr = 18h, Shadow = 000, Normal)

Note: To read MII Register 18h, Shadow Register *xxx*:

1. Write register 18h Shadow Register 111.
   Bit 15 = 0.
   Bits [14:12] = *xxx* (Shadow Register value).
   Bits [2:0] = 111 (Miscellaneous Control).
2. Read Register 18h.
   Data read is the value from Shadow Register *xxx*.
   Bits [2:0] = *xxx* (Shadow Register value).

To write MII Register 18h, Shadow Register *yyy*:

1. Write Register 18h.
   Bits [15:3] = *data* to write.
   Bits [2:0] = *yyy* (Shadow Register value).

Table 655: Auxiliary Control Register (PHY_Addr = 0x1, Reg_Addr = 18h, Shadow = 000, Normal)

Bit	Field	Description	Init	Access
15	External Loopback	When bit 15 is a 1, external loopback operation is enabled. A special cable must be inserted to provide a looped signal path. When the bit is 0, normal operation resumes. • 1 = External Loopback Enabled. • 0 = Normal Operation.	0	R/W
14	Extended Packet Length	When bit 14 of the Auxiliary Control Register is written to 1, the BCM57XX receives packets up to 25 KB in length. When the bit is written to 0, the BCM57XX only receives packets up to 4.5 KB in length. • 1 = Allow reception of extended length packets. • 0 = Allow normal length Ethernet packets only.	0	R/W
13:12	Edge Rate Control (1000BASE-T)	Bits 13 and 12 of the Auxiliary Control Register control the edge rate of the transmit DAC output waveform. • 00 = 4 ns. • 01 = 5 ns. • 10 = 3 ns. • 11 = 0 ns.	00	R/W
11	Reserved	Write as 0, ignore on read.	0	R/W
10	Transmit mode	Bit 10 of the Auxiliary Control Register must always be written to 1 for normal PHY operation. • 1 = Normal operation. • 0 = Test mode.	1	R/W
9:8	Reserved	Write as 0, ignore on read.	0	R/W
7	Disable Partial Response Filter	When bit 7 of the Auxiliary Control Register is written to 1, the transmitter partial response filter is disabled. When the bit is written to 0, the transmitter partial response filter is enabled. • 1 = Transmitter partial response filter disabled. • 0 = Transmitter partial response filter enabled.	0	R/W
6:4	Reserved	Write as 0, ignore on read.	0	R/W
5:4	Edge Rate Control (100BASE-TX) (BCM5705, BCM5721, and BCM5751 only)	Bits 5 and 4 of the Auxiliary Control Register control the edge rate of the 100BASE-TX transmit DAC output waveform: • 00 = 4 ns. • 01 = 5 ns. • 10 = 3 ns. • 11 = 0 ns.	00	R/W
	Reserved (Other devices)	Write as 0, ignore on read.	0	R/W
3	Reserved (BCM5705 only)	Write as 0, ignore on read.	0	R/W
	Diagnostic Mode (Other devices)	When bit 3 of the Auxiliary Control Register is written to 1, the BCM57XX enters a special mode to diagnose faults within the cable plant. See Application Notes for details. • 1 = Cable diagnostic mode enabled. • 0 = Normal operation.	0	R/W

Table 655: Auxiliary Control Register (PHY_Addr = 0x1, Reg_Addr = 18h, Shadow = 000, Normal) (Cont.)

Bit	Field	Description	Init	Access
2:0	Shadow Register Select	The Auxiliary Control Register provides access to eight registers using a shadow technique. These three bits written define which set of 13 upper bits are used. No setup is required. Register reads are determined by the previous write operation**.** • 000 = Normal Operation. • 001 = 10 BASE-T Register (see "Auxiliary Control Register (PHY_Addr = 0x1, Reg_Addr = 18h, Shadow = 001, 10BASE-T)" on page 648). • 010 = Power Control Register (see "Auxiliary Control Register (PHY_Addr = 0x1, Reg_Addr = 18h, Shadow = 010, Power Control)" on page 651). • 011 = Reserved. • 100 = Misc Test Register 1 (see "Auxiliary Control Register (PHY_Addr = 0x1, Reg_Addr = 18h, Shadow = 100, Misc Test 1)" on page 653). • 101 = Misc Test Register 2. • 110 = Reserved. • 111 = Misc Control Register (see "Auxiliary Control Register (PHY_Addr = 0x1, Reg_Addr = 18h, Shadow = 111, Misc Control)" on page 655).	000	R/W

Auxiliary Control Register (PHY_Addr = 0x1, Reg_Addr = 18h, Shadow = 001, 10BASE-T)

Table 656: Auxiliary Control Register (PHY_Addr = 0x1, Reg_Addr = 18h, Shadow = 001, 10BASE-T)

Bit	*Field*	*Description*	*Init*	*Access*
15	Manchester Code Error	Indicates that a Manchester code violation was received. This bit is valid only during 10BASE-T operation. • 1 = Manchester Code Error (10BASE-T). • 0 = No Manchester Code error.	0	R/O LH
14	EOF Error	Indicates that the End Of Frame (EOF) sequence was improperly received, or not received at all. This bit is valid only during 10BASE-T operation. • 1 = EOF error detected (10BASE-T). • 0 = No EOF error detected.	0	R/O LH
13	Polarity Error	Indicates that an analog input polarity error has been detected and corrected. This bit is valid only during 10BASE-T operation. • 1 = Channel polarity inverted. • 0 = Channel polarity correct.	0	R/O
12	Block RX_DV Extension (IPG)	When this bit is set, blocking of RX_DV signal is extended for four additional RXC cycles to extend IPG. • 1 = Block RX_DV for four additional RXC cycles for IPG. • 0 = Normal operation.	0	R/W
11	10BASE-T TXC Invert Mode	When set to 1, this bit causes the polarity of the 10BASE-T transmit clock to be inverted. Writing 0 restores normal transmit clock polarity. This bit is valid only during 10BASE-T operation. • 1 = Invert TXC output. • 0 = Normal operation.	0	R/W
10	Reserved	Write as 0, ignore on read.	0	R/O
9	Jabber Disable	Writing a 1 to bit 9 of the Auxiliary Control Register allows the user to disable the Jabber Detect function defined in the IEEE standard. This function shuts off the transmitter when a transmission request has exceeded a maximum time limit. Writing a 0 to this bit or resetting the chip restores normal operation. Reading this bit returns the value of Jabber Detect Disable. Valid for 10BASE-T operation only. • 1 = Jabber function disabled. • 0 = Jabber function enabled.	0	R/W

Table 656: Auxiliary Control Register (PHY_Addr = 0x1, Reg_Addr = 18h, Shadow = 001, 10BASE-T) (Cont.)

Bit	Field	Description	Init	Access
8:7	8—1000BASE-T Signal Detect Threshold (BCM5705, BCM5721, and BCM5751 only)	• 1 = Low SD threshold. • 0 = Normal SD threshold.	0	R/W
	7—10BASE-T Signal Detect Threshold (BCM5705, BCM5721, and BCM5751 only)	• 1 = Low SD threshold. • 0 = Normal SD threshold.	0	R/W
	8:7—HSQ:LSQ (Other devices)	Extends or decreases the squelch levels for detection of incoming 10BASE-T data packets. The default squelch levels implemented are those defined in the IEEE standard. The high- and low-squelch levels are useful for situations where the IEEE-prescribed levels are inadequate. The squelch levels are used by the CRS/LINK block to filter out noise and recognize only valid packet preambles and link integrity pulses. Using low squelch levels allows the BCM57XX to operate properly over longer cable lengths. Using high squelch levels can be useful in situations where there is a high level of noise present on the cables. Reading these two bits returns the value of the squelch levels. • 00 = Normal squelch. • 01 = Low squelch. • 10 = High squelch. • 11 = Reserved.	00	R/W
6	10BASE-T Echo Mode	When enabled, during 10BASE-T half-duplex transmit operation, the transmitted data is replicated on the receive data pins and the TXEN signal echoes onto the RXDV pin. The TXEN signal also echoes onto the CRS pin and the CRS deassertion directly follows the TXEN deassertion. • 1 = Echo transmit data to receive data. • 0 = Normal operation.	0	R/W
5	SQE Enable Mode	Writing a 1 to this bit enables SQE mode. Writing a 0 disables it.This bit is valid only during 10BASE-T operation. • 1 = Enable SQE. • 0 = Disable SQE.	0	R/W
4	10BASE-T No Dribble	When enabled, the PHY rounds down to the nearest nibble when dribble bits are present on the 10BASE-T input stream. • 1 = Correct 10BASE-T dribble nibble. • 0 = Normal operation.	0	R/W
3	Reserved(BCM5705, BCM5721, and BCM5751 only)		0	R/O
	Serial Mode 10BASE-T (Other devices)	When this bit is set ,10BASE-T serial mode is enabled in the MII side in 10BASE-T mode. • 1 = Enable 10BASE-T Serial Mode. • 0 = Normal operation.	0	R/W

Table 656: Auxiliary Control Register (PHY_Addr = 0x1, Reg_Addr = 18h, Shadow = 001, 10BASE-T) (Cont.)

Bit	Field	Description	Init	Access
2:0	Shadow Register Select	The Auxiliary Control Register provides access to eight registers using a shadow technique. These three bits written define which set of 13 upper bits is used. No setup is required. Register reads are determined by the previous write operation. • 000 = Normal Operation (see "Auxiliary Control Register (PHY_Addr = 0x1, Reg_Addr = 18h, Shadow = 000, Normal)" on page 645). • 001 = 10 BASE-T Register. • 010 = Power Control Register (see "Auxiliary Control Register (PHY_Addr = 0x1, Reg_Addr = 18h, Shadow = 010, Power Control)" on page 651). • 011 = Reserved. • 100 = Misc Test Register 1 (see "Auxiliary Control Register (PHY_Addr = 0x1, Reg_Addr = 18h, Shadow = 100, Misc Test 1)" on page 653). • 101 = Misc Test Register 2. • 110 = Reserved. • 111 = Misc Control Register (see "Auxiliary Control Register (PHY_Addr = 0x1, Reg_Addr = 18h, Shadow = 111, Misc Control)" on page 655).	000	R/W

Auxiliary Control Register (PHY_Addr = 0x1, Reg_Addr = 18h, Shadow = 010, Power Control)

BCM5705, BCM5721, and BCM5751 MAC Transceivers Only

This version of the Auxiliary Control Register (PHY_Addr = 0x1, Reg_Addr = 18h, Shadow = 010, Power Control) applies to the BCM5705, BCM5721, and BCM5751 MAC Transceivers only.

Table 657: Aux. Reg. (PHY_Addr=0x1, Reg_Addr=18h, Shadow=010, Pwr Cont., 5705/5721/5751 only)

Bit	*Field*	*Description*	*Init*	*Access*
15:6	Reserved	-	00Eh	R/W
5	Super Isolate	N/A	0	R/W
4:3	Reserved	Write as 1, ignore on read.	10	R/W
2:0	Shadow Register Select	The Auxiliary Control Register provides access to eight registers using a shadow technique. These three bits written define which set of 13 upper bits is used. No setup is required. Register reads are determined by the previous write operation. • 000 = Normal Operation (see "Auxiliary Control Register (PHY_Addr = 0x1, Reg_Addr = 18h, Shadow = 000, Normal)" on page 645). • 001 = 10 BASE-T Register (see "Auxiliary Control Register (PHY_Addr = 0x1, Reg_Addr = 18h, Shadow = 001, 10BASE-T)" on page 648). • 010 = Power Control Register. • 011 = Reserved. • 100 = Misc Test Register 1 (see "Auxiliary Control Register (PHY_Addr = 0x1, Reg_Addr = 18h, Shadow = 100, Misc Test 1)" on page 653). • 101 = Misc Test Register 2. • 110 = Reserved. • 111 = Misc Control Register (see "Auxiliary Control Register (PHY_Addr = 0x1, Reg_Addr = 18h, Shadow = 111, Misc Control)" on page 655).	000	R/W

Rest of BCM57XX Family

This version of the Auxiliary Control Register (PHY_Addr = 0x1, Reg_Addr = 18h, Shadow = 010, Power Control) applies to the rest of the BCM57XX family.

Table 658: Aux. Cont. (PHY_Addr=0x1, Reg_Addr=18h, Shadow=010, Pwr Cont., Other BCM57XX Fam.)

Bit	Field	Description	Init	Access
15:14	Reserved	Write as 1, ignore on read.	0000	R/W
13	Regulator input voltage select	If the regulator input supply is 3.3V, set this bit to a 1. If the regulator input supply voltage is 2.5V, then set this bit to a 0. • 1 = Regulator input voltage, REGSUP = 3.3V. • 0 = Regulator input voltage, REGSUP = 2.5V.	0	R/W
12	Regulator output voltage select	The regulator output voltage can be set to 1.8V, 1.5V or 1.3V. Bit 10 and 9 determines the regulator output voltage. • 00 = Regulator output voltage should be 1.8V. • 01 = Regulator output voltage should be 1.5V. • 10 = Regulator output voltage should be 1.3V.	00	R/W
11:6	Reserved	-		
5	Super Isolate	• 1 = isolate mode with no link pulses transmitted • 0 = normal operation	0	R/W
4	Reserved	Write as 1, ignore on read.	1	R/W
3	Wake on LAN	Writing a 1 to this bit enables the Wake on LAN capability of the transceiver. Writing a 0 disables Wake on LAN. • 1 = Enable Wake on LAN operation. • 0 = Normal operation.	0	R/W
2:0	Shadow Register Select	The Auxiliary Control Register provides access to eight registers using a shadow technique. These three bits written define which set of 13 upper bits is used. No setup is required. Register reads are determined by the previous write operation. • 000 = Normal Operation (see "Auxiliary Control Register (PHY_Addr = 0x1, Reg_Addr = 18h, Shadow = 000, Normal)" on page 645). • 001 = 10 BASE-T Register (see "Auxiliary Control Register (PHY_Addr = 0x1, Reg_Addr = 18h, Shadow = 001, 10BASE-T)" on page 648). • 010 = Power Control Register. • 011 = Reserved. • 100 = Misc Test Register 1 (see "Auxiliary Control Register (PHY_Addr = 0x1, Reg_Addr = 18h, Shadow = 100, Misc Test 1)" on page 653). • 101 = Misc Test Register 2. • 110 = Reserved. • 111 = Misc Control Register (see "Auxiliary Control Register (PHY_Addr = 0x1, Reg_Addr = 18h, Shadow = 111, Misc Control)" on page 655).	000	R/W

Auxiliary Control Register (PHY_Addr = 0x1, Reg_Addr = 18h, Shadow = 100, Misc Test 1)

Table 659: Auxiliary Control Register (PHY_Addr = 0x1, Reg_Addr = 18h, Shadow = 100, Misc Test 1)

Bit	Field	Description	Init	Access
15	Lineside [Remote] Loopback Enable (BCM5705, BCM5721, and BCM5751 only)	Setting bit 15 enables lineside [remote] loopback of the copper receive packet back out through the MDI transmit path. • 1 = Enable lineside [remote] loopback from MDI (cable end) receive packet, through PCS and back to MDI transmit packet. • 0 = Disable loopback.	0	R/W
	Reserved	Write as 0, ignore on read.	0	R/O
14:12	Reserved	-	000	R/O
11	Lineside [Remote] Loopback Tri-state (BCM5705, BCM5721, and BCM5751 only)	Setting bit 11 tri-states the receive MII pins when the device is in lineside [remote] loopback mode. • 1 = Tri-state the receive MII pins when lineside [remote] loopback is enabled. • 0 = Lineside [remote] loopback packets appear on MII.		
10:5	Reserved	Write as 0, ignore on read.	00h	R/O
4	Swap RX MDIX	When this bit is set to 1, the transmitter and receiver operate on the same twisted-pair. This function is for use in a test mode in which the transmitter output is detected by the receiver attached to the same pair. • 1 = RX and TX operate on same pair. • 0 = Normal operation.	0	R/O
3	TXHalfout	Setting this bit to 1 reduces the output of the transmitter to half of its normal amplitude when operating in 10 Mbit or 100BASE-TX mode. Setting it to 0 restores full amplitude operation. This function is for use in a test mode in which an unterminated output delivers a reflected signal with twice the amplitude of a terminated output. • 1 = Transmit 10BASE-T and 100BASE-T at half amplitude. • 0 = Normal operation.	0	R/W

Table 659: Auxiliary Control Register (PHY_Addr = 0x1, Reg_Addr = 18h, Shadow = 100, Misc Test 1) (Cont.)

Bit	*Field*	*Description*	*Init*	*Access*
2:0	Shadow Register Select	The Auxiliary Control Register provides access to eight registers using a shadow technique. These three bits written define which set of 13 upper bits is used. No setup is required. Register reads are determined by the previous write operation. • 000 = Normal Operation (see "Auxiliary Control Register (PHY_Addr = 0x1, Reg_Addr = 18h, Shadow = 000, Normal)" on page 645). • 001 = 10 BASE-T Register (see "Auxiliary Control Register (PHY_Addr = 0x1, Reg_Addr = 18h, Shadow = 001, 10BASE-T)" on page 648). • 010 = Power Control Register (see "Auxiliary Control Register (PHY_Addr = 0x1, Reg_Addr = 18h, Shadow = 010, Power Control)" on page 651). • 011 = Reserved. • 100 = Misc Test Register 1. • 101 = Misc Test Register 2. • 110 = Reserved. • 111 = Misc Control Register (see "Auxiliary Control Register (PHY_Addr = 0x1, Reg_Addr = 18h, Shadow = 111, Misc Control)" on page 655).	000	R/W

Auxiliary Control Register (PHY_Addr = 0x1, Reg_Addr = 18h, Shadow = 111, Misc Control)

Table 660: Auxiliary Control Register (PHY_Addr = 0x1, Reg_Addr = 18h, Shadow = 111, Misc Control)

Bit	Field	Description	Init	Access
15	Write Enable (Bits 11:3) (BCM5705, BCM5721, and BCM5751 only)	• 1 = Write bits 14:0. • 0 = Write bits 14:12 and 2:0.	0	R/W SC
	Write Enable (Bits 8:3) (Other devices)	• 1 = Write bits 8:3. • 0 = Only write bits 14:12.	0	R/W SC
14:12	Shadow Register Read Selector	• 000 = Normal Operation (see "Auxiliary Control Register (PHY_Addr = 0x1, Reg_Addr = 18h, Shadow = 000, Normal)" on page 645). • 001 = 10 BASE-T Register (see "Auxiliary Control Register (PHY_Addr = 0x1, Reg_Addr = 18h, Shadow = 001, 10BASE-T)" on page 648). • 010 = Power Control Register (see "Auxiliary Control Register (PHY_Addr = 0x1, Reg_Addr = 18h, Shadow = 010, Power Control)" on page 651). • 011 = Reserved. • 100 = Misc Test Register 1 (see "Auxiliary Control Register (PHY_Addr = 0x1, Reg_Addr = 18h, Shadow = 100, Misc Test 1)" on page 653). • 101 = Misc Test Register 2. • 110 = Reserved. • 111 = Misc Control Register.	000	R/W
11:9	11—Packet Counter Mode (BCM5705, BCM5721, and BCM5751 only)	• 1 = Receive packet counter. • 0 = Transmit packet counter.	0	R/W
	10—Reserved (BCM5705, BCM5721, and BCM5751 only)	-	0	R/W
	9—Force Auto-MDIX Mode (BCM5705, BCM5721, and BCM5751 only)	• 1 = Auto-MDIX is enabled when auto-negotiation is disabled. • 0 = Auto-MDIX is disabled when auto-negotiation is disabled.		
	11:9—Cable Length 100BASE-TX (Other devices)	3 bit cable length for 100BASE-TX. Invalid during 100BASE-TX WOL.	000	R/O
8	RGMII Timing Mode (BCM5705, BCM5721, and BCM5751 only)	• 1 = RGMII RXC delayed timing mode. • 0 = RGMII RXC/RXD aligned timing mode.	0	R/W
	100BASE-TX High Performance (Other devices)	• 1 = Use Gigabit Equivalizer for High Performance 100BASE-TX. • 0 = Normal operation.	0	R/W
7	RGMII Mode	N/A	0	R/W
6	RGMII RXER Mode	N/A	0	R/W
5	RGMII Out Of Band Status Disable	• 1 = Send regular data during IPG. • 0 = Send Out-Of-Band Status info in RGMII mode.	0	R/W

Table 660: Auxiliary Control Register (PHY_Addr = 0x1, Reg_Addr = 18h, Shadow = 111, Misc Control) (Cont.)

Bit	Field	Description	Init	Access
4	Reserved (BCM5705, BCM5721, and BCM5751 only)		1	R/W
	Wire Speed Enable	• 1 = Enable Wire Speed mode. • 0 = Normal operation.	0	R/W
3	MDIO All PHY Select	• 1 = All PHY selected during MDIO writes when PHY address = 00000b. • 0 = Normal operation.	0	R/W
2:0	Shadow Register Select	The Auxiliary Control Register provides access to eight registers using a shadow technique. These three bits written define which set of 13 upper bits are used. No setup is required. Register reads are determined by the previous write operation. • 000 = Normal Operation (see "Auxiliary Control Register (PHY_Addr = 0x1, Reg_Addr = 18h, Shadow = 000, Normal)" on page 645). • 001 = 10 BASE-T Register (see "Auxiliary Control Register (PHY_Addr = 0x1, Reg_Addr = 18h, Shadow = 001, 10BASE-T)" on page 648). • 010 = Power Control Register (see "Auxiliary Control Register (PHY_Addr = 0x1, Reg_Addr = 18h, Shadow = 010, Power Control)" on page 651). • 011 = Reserved. • 100 = Misc Test Register 1 (see "Auxiliary Control Register (PHY_Addr = 0x1, Reg_Addr = 18h, Shadow = 100, Misc Test 1)" on page 653). • 101 = Misc Test Register 2. • 110 = Reserved. • 111 = Misc Control Register.	000	R/W

AUXILIARY STATUS SUMMARY REGISTER (PHY_ADDR = 0X1, REG_ADDR = 19H)

Table 661: Auxiliary Status Summary Register (PHY_Addr = 0x1, Reg_Addr = 19h)

Bit	*Field*	*Description*	*Init*	*Access*
15	Auto-negotiation Complete	The BCM57XX returns a 1 on bit 15 of the Auxiliary Status Summary Register when auto-negotiation is complete. This bit returns a 0 while auto-negotiation is in progress. • 1 = Auto-negotiation complete. • 0 = Auto-negotiation in progress.	0	R/O
14	Auto-negotiation Complete Acknowledge	The BCM57XX returns a 1 on bit 14 of the Auxiliary Status Summary Register 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. • 1 = Entered auto-negotiation link good check state. • 0 = State not entered since last read.	0	R/O LH
13	Auto-negotiation Acknowledge Detect	The BCM57XX returns a 1 on bit 13 of the Auxiliary Status Summary Register 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. • 1 = Entered auto-negotiation acknowledge detect state. • 0 = State not entered since last read.	0	R/O LH
12	Auto-negotiation Ability Detect	The BCM57XX returns a 1 on bit 12 of the Auxiliary Status Summary Register 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. • 1 = Entered auto-negotiation ability detect state. • 0 = State not entered since last read.	0	R/O LH
11	Auto-negotiation Next Page Wait	The BCM57XX returns a 1 on bit 11 of the Auxiliary Status Summary Register 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. • 1 = Entered auto-negotiation next page wait state. • 0 = State not entered since last read.	0	R/O LH
10:8	Auto-negotiation HCD (Current Operating Speed and Duplex Mode)	Bits 10:8 of the Auxiliary Status Summary Register report the mode of operation negotiated between the BCM57XX and its link partner. The bits return 000 until auto-negotiation has completed, as reported by bit 15 of the auxiliary status summary register. When the auto-negotiation function has been disabled, bits 10:8 report the manually selected mode of operation. • 111 = 1000BASE-T full-duplex* • 110 = 1000BASE-T half-duplex* • 101 = 100BASE-TX full-duplex* • 100 = 100BASE-T4 • 011 = 100BASE-TX half-duplex* • 010 = 10BASE-T full-duplex* • 001 = 10BASE-T half-duplex* • 000 = No highest common denominator or auto-negotiation not complete.	000	R/O

Table 661: Auxiliary Status Summary Register (PHY_Addr = 0x1, Reg_Addr = 19h) (Cont.)

Bit	***Field***	***Description***	***Init***	***Access***
7	Parallel Detection Fault	Bit 7 of the Auxiliary Status Summary Register returns a one when a parallel detection fault has occurred in the auto-negotiation state machine. When a parallel detection fault occurs, this bit is latched at one and remain so until the register read. This bit returns a 0 when a parallel detection fault has not occurred since the last time it was read. • 1 = Parallel link fault detected. • 0 = Parallel link fault not detected.	0	R/O LH
6	Remote Fault	The BCM57XX returns a one on bit 6 of the Auxiliary Status Summary Register when the link partner has advertised detection of a remote fault, otherwise, it returns a 0. • 1 = Link partner has detected remote fault. • 0 = Link partner has not detected remote fault.	0	R/O
5	Auto-negotiation Page Received	The BCM57XX returns a one on bit 5 of the Auxiliary Status Summary Register 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. • 1 = New page has been received from link partner. • 0 = New page has not been received.	0	R/O LH
4	Link Partner Auto-negotiation Ability	The BCM57XX returns a one on bit 4 of the Auxiliary Status Summary Register 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. • 1 = Link partner has auto-negotiation capability. • 0 = Link partner does not perform auto-negotiation.	0	R/O
3	Link Partner Next Page Ability	The BCM57XX returns a one on bit 3 of the Auxiliary Status Summary Register when the link partner needs to transmit Next Page information, otherwise, it returns a 0. • 1 = Link partner has Next Page capability. • 0 = Link partner does not have Next Page capability.	0	R/O
2	Link Status	The BCM57XX returns a 1 on bit 2 of the auxiliary status summary register when the link status is good, otherwise, it returns a 0. • 1 = Link is up (Link Pass state). • 0 = Link is down (Link Fail state).	0	R/O
1	Pause Resolution–Receive Direction	When auto-negotiation has completed, the BCM57XX returns the result of the pause resolution function for full-duplex flow control on bits 1:0 of the auxiliary status summary register. When bit 1 returns a 1, the link partner can send pause frames toward the local device. These bits are only guaranteed to be valid when bit 15 of the Auxiliary Status Summary Register is 1. • 1 = Enable pause receive. • 0 = Disable pause receive.	0	R/O

Table 661: Auxiliary Status Summary Register (PHY_Addr = 0x1, Reg_Addr = 19h) (Cont.)

Bit	Field	Description	Init	Access
0	Pause Resolution–Transmit Direction	When auto-negotiation has completed, the BCM57XX returns the result of the pause resolution function for full-duplex flow control on bits 1:0 of the auxiliary status summary register. 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. • 1 = Enable pause transmit. • 0 = Disable pause transmit.	0	R/O
* Indicates the negotiated HCD when auto-negotiation enable = 1. Indicates the manually selected speed and duplex mode when auto-negotiation enable = 0.				

INTERRUPT STATUS REGISTER (PHY_ADDR = 0X1, REG_ADDR = 1AH)

The interrupt status output is asserted when any bit in this register is set and the corresponding bit in the interrupt mask register is cleared.

Table 662: Interrupt Status Register (PHY_Addr = 0x1, Reg_Addr = 1Ah)

Bit	Field	Description	Init	Access
15	Signal Detect/Energy Detect Change (BCM5705, BCM5721, BCM5751, BCM5714, and BCM5715 only)	The bit indicates whether the fiber SD or the copper ED has changed since the last read. • 1 = SD or ED changed (enabled by register 1Ch, shadow 00101, bit 5 =1). • 0 = Interrupt cleared.	0	R/O
	Reserved (other devices)	Ignore on read.	0	R/O
14	Illegal Pair Swap	The BCM57XX returns a one on bit 14 of the Interrupt Status Register 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. • 1 = Illegal pair swap detected. • 0 = Interrupt cleared.	0	R/O LH
13	MDIX Status Change	The BCM57XX returns a one on bit 13 of the Interrupt Status Register 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. • 1 = MDIX status changed since last read. • 0 = Interrupt cleared.	0	R/O LH
12	Exceeded High Counter Threshold	The BCM57XX returns a one on bit 12 of the Interrupt Status Register when one or more of the counters in registers 12-14h is above 32K, otherwise, it returns a 0. • 1 = Value in one or more counters is above 32K. • 0 = All counters below 32K.	0	R/O

Table 662: Interrupt Status Register (PHY_Addr = 0x1, Reg_Addr = 1Ah) (Cont.)

Bit	Field	Description	Init	Access
11	Exceeded Low Counter Threshold	The BCM57XX returns a one on bit 11 of the Interrupt Status Register when one or more of the counters in registers 12-14h is above 128, otherwise, it returns a 0. • 1 = Value in one or more counters is above 128. • 0 = All counters below 128.	0	R/O
10	Auto-negotiation Page Received	The BCM57XX returns a one on bit 10 of the Interrupt Status Register 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. • 1 = Page received since last read. • 0 = Interrupt cleared.	0	R/O LH
9	HCD No Link	Bit 9 of the Interrupt Status Register is set to 1 by the BCM57XX when the negotiated HCD was not able to establish a link. The bit is cleared when the register is read. • 1 = Negotiated HCD, did not establish link. • 0 = Interrupt cleared.	0	R/O LH
8	No HCD	Bit 8 of the Interrupt Status Register is set to 1 by the BCM57XX when auto-negotiation returns no HCD. The bit is cleared when the register is read. • 1 = Auto-negotiation returned HCD = none. • 0 = Interrupt cleared.	0	R/O LH
7	Negotiated Unsupported HCD	Bit 7 of the Interrupt Status Register is set to 1 when the auto-negotiation HCD is not supported by the BCM57XX. The BCM57XX does not support 100BASE-T4. The bit is cleared when the register is read. • 1 = Auto-negotiation HCD not supported by BCM57XX. • 0 = Interrupt cleared.	0	R/O LH
6	Scrambler Synchronization Error	he BCM57XX returns a one on bit 6 of the Interrupt Status Register when a scrambler synchronization error has been detected since the last time this register was read, otherwise, it returns a 0. • 1 = Scrambler synchronization error occurred since last read. • 0 = Interrupt cleared.	0	R/O LH
5	Remote Receiver Status Change	The BCM57XX returns a one on bit 5 of the Interrupt Status Register when the remote receiver status has changed since the last time this register was read, otherwise, it returns a 0. • 1 = Remote receiver status changed since last read. • 0 = Interrupt cleared.	0	R/O LH
4	Local Receiver Status Change	The BCM57XX returns a one on bit 4 of the Interrupt Status Register when the local receiver status has changed since the last time this register was read, otherwise, it returns a 0. • 1 = Local receiver status changed since last read. • 0 = Interrupt cleared.	0	R/O LH
3	Duplex Mode Change	The BCM57XX returns a one on bit 3 of the Interrupt Status Register when the duplex mode has changed since the last time this register was read, otherwise, it returns a 0. • 1 = Duplex mode changed since last read. • 0 = Interrupt cleared.	0	R/O LH

Table 662: Interrupt Status Register (PHY_Addr = 0x1, Reg_Addr = 1Ah) (Cont.)

Bit	Field	Description	Init	Access
2	Link Speed Change	The BCM57XX returns a one on bit 2 of the Interrupt Status Register when the link speed has changed since the last time this register was read, otherwise, it returns a 0. • 1 = Link speed changed since last read. • 0 = Interrupt cleared.	0	R/O LH
1	Link Status Change	The BCM57XX returns a one on bit 1 of the Interrupt Status Register when the link status has changed since the last time this register was read, otherwise, it returns a 0. • 1 = Link status changed since last read. • 0 = Interrupt cleared.	0	R/O LH
0	CRC Error	The BCM57XX returns a one on bit 0 of the Interrupt Status Register when a receive CRC error has been detected since the last time this register was read, otherwise, it returns a 0. • 1 = CRC error occurred since last read. • 0 = Interrupt cleared.	0	R/O LH

INTERRUPT MASK REGISTER (PHY_ADDR = 0X1, REG_ADDR = 1BH)

Table 663: Interrupt Mask Register (PHY_Addr = 0x1, Reg_Addr = 1Bh)

Bit	Field	Description	Init	Access
15:0	Interrupt Mask Vector	When bit n of the 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. • 1 = Interrupt masked, status bits operate normally. • 0 = Interrupt enabled, status bits operate normally.	11…1	R/W

Misc Shadow Registers (PHY_ADDR = 0X1, REG_ADDR = 1CH; BCM5702, BCM5703, and BCM5704 Only)

Spare Control Register 1 (Address 1Ch, Enable by Register 1Ch Bits[14:10] = 00010)

This version of the register is only applicable to the BCM5702, BCM5703, and BCM5704.

Table 664: Spare Control Register 1 (Address 1Ch, Enable by Register 1Ch Bits[14:10] = 00010)

Bit	Field	Description	Init	Access
15	Write Enable	• 1 = Write bits [9:0]. • 0 = Read bits [9:0].	0	R/W
14:10	Shadow Register Selector	• 00010 = Spare Control Register 1. • 00100 = Spare Control Register 2. • 00101 = Spare Control Register 3. • 01000 = LED Status Register. • 01001 = LED Control Register. • 01010 = Auto Power Down Register.	00010	R/W
9:2	Reserved	Write as 00000000, Ignore when read.	00h	R/W
1	Ultra Low Power Mode	• 1 = Ultra Low-power Mode Enabled. • 0 = Normal Mode.	0	R/W
0	Link LED Mode	• 1 = Alternative Link LED mode. • 0 = Normal Link LED mode.	0	R/W

Write Enable

Setting bit 15 to a 1 during a write to this register 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 the desired shadow values in bits [14:10]. The next MDIO read of Register address 1Ch will contain the desired shadow register values in bits [9:0].

Shadow Register Selector

Register Address 1Ch bits [14:10] provides access to 4 registers using a shadow technique. The bits define which set of the lower 10 bits is used in accordance with the table description for the Shadow Register Selector. The register set shown above is that for Spare Control Register 1, obtained when bits [14:10] = 00010.

Ultra Low-Power Mode

Writing a 1 to bit 1 will put the BCM57XX into Ultra Low-Power Mode. There is no register access once this mode is entered. To get out of this mode, a hardware reset is required.

Link LED Mode

By writing a 1 to this bit, the BCM57XX operate in an alternative Link/Speed LED scheme as follows: $\overline{\text{LINK[1]}}$ and $\overline{\text{LINK[2]}}$ pins will function as Speed indicators only, and $\overline{\text{SLAVE}}$ pin will function as a Link LED.

Spare Control Register 2 (Address 1Ch, Shadow Value 00100)

This register is only applicable to the BCM5702, BCM5703, and BCM5704.

Table 665: Spare Control Register 2 (Address 1Ch, Shadow Value 00100)

Bit	Field	Description	Init	Access
15	Write Enable	• 1 = Write bits [9:0]. • 0 = Read bits [9:0].	0	R/W
14:10	Shadow Register Selector	• 00010 = Spare Control Register 1. • 00100 = Spare Control Register 2. • 00101 = Spare Control Register 3. • 01000 = LED Status Register. • 01001 = LED Control Register. • 01010 = Auto Power Down Register.	00100	R/W
9:7	Reserved	Write as 000, Ignore when read.	000	R/W
6	Invert TBI	• 1 = Value latched in during power-on/reset on EN_10B pin, inverted. • 0 = Value latched in during power-on/reset on EN_10B pin, not inverted.	0	R/W
5	Reserved	Write as 0, Ignore when read.	0	R/W
4	Invert $\overline{\text{INTR}}$	• 1 = $\overline{\text{INTR}}$ pin active high. • 0 = $\overline{\text{INTR}}$ pin active low.	0	R/W
3:2	Reserved	Write as 00, Ignore when read.	00	R/W
1	Energy Detect function	• 1 = Energy Detect function enabled. • 0 = Energy Detect function disabled.	0	R/W
0	Reserved	Write as 0, Ignore when read.	0	R/W

Write Enable

Setting bit 15 to a 1 during a write to this register 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 the desired shadow values in bits [14:10]. The next MDIO read of Register address 1Ch will contain the desired shadow register values in bits [9:0].

Shadow Register Selector

Register Address 1Ch bits [14:10] provide access to four registers using a shadow technique. The bits define which set of the lower 10 bits is used in accordance with the table description for the Shadow Register Selector. The register set shown above is that for Spare Control Register 1, obtained when bits [14:10] = 00010.

Invert TBI

Writing a 1 to the Spare Control Register 2, Bit 6, toggles the value latched in on the EN_10B during reset as follows:

- If EN_10B pin = 0 and RGMIIEN pin = 0 at reset, then setting bit 6 = 1 will enable TBI mode, enable RBC0/1 and tri-state RXC/TXC.
- If EN_10B pin = 1 and RGMIIEN pin = 0 at reset, then setting bit 6 = 1 will disable TBI mode, enable RXC/TXC and tri-state RBC0/1.
- If EN_10B pin = 0 and RGMIIEN pin = 1 at reset, then setting bit 6 = 1 will enable RTBI mode, enable RXC/TXC and RBC0/1.

- If EN_10B pin = 1 and RGMIIEN pin = 1 at reset, then setting bit 6 = 1 will disable RTBI mode, enable RXC/TXC and RBC0/1.

Invert $\overline{\text{INTR}}$

When bit 4 of the Spare Control Register 2 is written to 1, the $\overline{\text{INTR}}$ pin will become an active high output. When bit 4 is written to 0, the $\overline{\text{INTR}}$ pin will become an active low output.

Energy Detect

Writing a 1 to bit 1 of MII Register 1Ch with Shadow Value 00100 will enable the Energy Detect function only when auto-negotiation is also enabled. The Interrupt function will be unavailable while this bit is set.

Spare Control Register 3 (Address 1Ch, Shadow Value 00101)

This register is only applicable to the BCM5702, BCM5703, and BCM5704.

Table 666: Spare Control Register 3 (Address 1Ch, Shadow Value 00101)

Bit	Field	Description	Init	Access
15	Write Enable	• 1 = Write bits [9:0]. • 0 = Read bits [9:0].	0	R/W
14:10	Shadow Register Selector	• 00010 = Spare Control Register 1. • 00100 = Spare Control Register 2. • 00101 = Spare Control Register 3. • 01000 = LED Status Register. • 01001 = LED Control Register. • 01010 = Auto Power Down Register.	00101	R/W
9:5	Reserved	Write as 00000, Ignore when read.	00000	R/W
4:2	Reserved	Write as 111, Ignore on read.	111	R/W
1	CLK125 Enable	• 1 = Enables CLK125 output. • 0 = Disables CLK125 output.	1	R/W
0	Reserved	Write as 1, Ignore on read.	1	R/W

Write Enable

Setting bit 15 to a 1 during a write to this register 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 the desired shadow values in bits [14:10]. The next MDIO read of Register address 1Ch will contain the desired shadow register values in bits [9:0].

Shadow Register Selector

Register Address 1Ch bits [14:10] provide access to 4 registers using a shadow technique. The bits define which set of the lower 10 bits is used in accordance with the table description for the Shadow Register Selector. The register set shown above is that for Spare Control Register 1, obtained when bits [14:10] = 00010.

CLK125 Enable

Writing a 0 to this bit will disable the CLK125 output only when the part is in Auto Power Down Mode. This feature enables additional power savings. This feature should only be used if the 125-MHz clock is not needed by the MAC or ASIC.

LED Status Register (Address 1Ch, Shadow Register Selector = 01000)

This register is only applicable to the BCM5702, BCM5703, and BCM5704.

Table 667: LED Status Register (Address 1Ch, Shadow Register Selector = 01000)

Bit	Field	Description	Init	Access
15	Write Enable	• 1 = Write bits [9:0]. • 0 = Read bits [9:0].	0	R/W
14:10	Shadow Register Selector	• 00010 = Spare Control Register 1. • 00100 = Spare Control Register 2. • 00101 = Spare Control Register 3. • 01000 = LED Status Register. • 01001 = LED Control Register. • 01010 = Auto Power Down Register.	01000	R/W
9	Reserved	Write as 0, ignore when read.	0	R/O
8	SLAVE	• 1 = Master mode. • 0 = Slave mode.	0	R/O
7	FDX	• 1 = HDX mode. • 0 = FDX mode.	0	R/O
6	INTR	• 1 = No active interrupts. • 0 = Active Interrupts.	0	R/O
5	Link1000	• 1 = No 1000BASE-T established. • 0 = 1000BASE-T established.	0	R/O
4	Link100	• 1 = No 100BASE-TX established. • 0 = 100BASE-TX established.	0	R/O
3	Link10	• 1 = No 10BASE-T established. • 0 = 10BASE-T established.	0	R/O
2	TRANSMIT	• 1 = Transmit not active. • 0 = Transmit active.	0	R/O
1	RECEIVE	• 1 = Receive not active. • 0 = Receive active.	0	R/O
0	QUALITY	• 1 = Quality not good. • 0 = Quality good.	0	R/O

Write Enable

Setting bit 15 to a 1 during a write to this register 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 the desired shadow values in bits [14:10]. The next MDIO read of Register address 1Ch will contain the desired shadow register values in bits [9:0].

Shadow Register Selector

Bits [14:10] of this register should be set to 01000 to enable read/write to the Spare Control Register, address 1Ch.

$\overline{\text{SLAVE}}$

When this bit returns a 0, the device is in the Slave mode. When this bit returns a 1, the device is in the Master mode.

$\overline{FDX}$

When this bit returns a 0, the device is in the full-duplex mode. When this bit returns a 1, the device is in the half-duplex mode.

$\overline{INTR}$

When this bit returns a 0, the device has active interrupts. When this bit returns a 1, the device is has no active interrupts.

$\overline{LINK1000}$

When this bit returns a 0, the device has established a 1000BASE-T link. When this bit returns a 1, the device has not established a link.

$\overline{LINK100}$

When this bit returns a 0, the device has established a 100BASE-TX link. When this bit returns a 1, the device has not established a link.

$\overline{LINK10}$

When this bit returns a 0, the device has established a 10BASE-T link. When this bit returns a 1, the device has not established a link.

$\overline{TRANSMIT}$

When this bit returns a 0, the device is transmitting data. When this bit returns a 1, the device is not transmitting data.

$\overline{RECEIVE}$

When this bit returns a 0, the device is receiving data. When this bit returns a 1, the device is not receiving data.

$\overline{QUALITY}$

When this bit returns a 0, the device has good signal Quality. When this bit returns a 1, the device does not have good signal Quality.

LED Control Register (Address 1Ch, Shadow Value 01001)

This register is only applicable to the BCM5702, BCM5703, and BCM5704.

Table 668: LED Control Register (Address 1Ch, Shadow Value 01001)

Bit	Field	Description	Init	Access
15	Write Enable	• 1 = Write bits [9:0]. • 0 = Read bits [9:0].	0	R/W
14:10	Shadow Register Selector	• 00010 = Spare Control Register 1. • 00100 = Spare Control Register 2. • 00101 = Spare Control Register 3. • 01000 = LED Status Register. • 01001 = LED Control Register. • 01010 = Auto Power Down Register.	01001	R/W
9:5	Reserved	Write as 0, ignore when read.	00h	R/W
4	Activity Link LED Enable	• 1 = Drive activity/link data on receive LED. • 0 = Normal operation.	0	R/W
3	Activity LED Enable	• 1 = Drive activity data on receive LED. • 0 = Normal operation.	1	R/W
2	Remote Fault LED Enable	• 1 = Drive remote fault on quality LED. • 0 = Normal operation.	0	R/W
1:0	Link Utilization LED Selector	• 00 = Normal operation. • 01 = Transmit data on receive LED. • 10 = Receive data on receive LED. • 11 = Activity data on receive LED. (This mode has higher priority than the activity LED enable in bit 3.)	00	R/W

Write Enable

During a write to this register, setting 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

LED Control Register bits [14:10] of this register must be set to *01001* to enable read/write to the Register address 1Ch.

Activity Link LED Enable

Setting LED Control Register bit 4 drives Activity/Link data on receive LED.

Activity LED Enable

Setting LED Control Register bit 3 drives activity data on receive LED.

Remote Fault LED Enable

Setting LED Control Register bit 2 drives remote fault on $\overline{\text{QUALITY}}$ LED.

Link Utilization LED Selector

Setting LED Control Register bits [1:0] drives link utilization on the $\overline{\text{RECEIVE}}$ LED.

Auto Power-Down Register (Address 1Ch, Shadow Value 01010)

This register is only applicable to the BCM5702, BCM5703, and BCM5704.

Table 669: Auto Power-Down Register (Address 1Ch, Shadow Value 01010)

Bit	Field	Description	Init	Access
15	Write Enable	• 1 = Write bits [9:0]. • 0 = Read bits [9:0].	0	R/W
14:10	Shadow Register Selector	• 00010 = Spare Control Register 1. • 00100 = Spare Control Register 2. • 00101 = Spare Control Register 3. • 01000 = LED Status Register. • 01001 = LED Control Register. • 01010 = Auto Power-Down Register.	01010	R/W
9:6	Reserved	Write as 0000, Ignore when read.	0000	R/W
5	Auto Power-Down Mode	• 1 = Auto Power-Down Mode Enabled. • 0 = Auto Power-Down Mode Disabled.	0	R/W
4	Sleep Timer	• 1 = Sleep timer is 5.4 seconds. • 0 = Sleep timer is 2.7 seconds.	0	
3:0	Wakeup Timer Select	Counter for Wakeup timer in units of 84 ms.	0001	

Write Enable

For reading the values of bits [9:0], write to Register 1Ch, Shadow 00010, bit 15 = 1 and the desired shadow values in bits [14:10]. The following MDIO read of Register address 1Ch will contain the desired shadow register values in bits [9:0].

Shadow Register Selector

Register Address 1Ch bits [14:10] provide access to four registers using a shadow technique. The bits define which set of the lower 10 bits are used in accordance with the table description for the Shadow Register Selector. The register set shown above is that for Spare Control Register 1, obtained when bits [14:10] = 00010.

Auto Power-Down Mode Enable

Setting this bit enables the Auto Power-Down Mode.

Sleep Timer

Setting this bit changes the wake-up time leaving Auto Power-Down Mode.

Wakeup Timer Select

The BCM57XX continues wake-up mode for a time based on the count stored in this register. The minimum value is 84 ms and the maximum value is 1.26s.

MISC SHADOW REGISTERS (PHY_ADDR = 0X1, REG_ADDR = 1CH; BCM5705, BCM5721, BCM5751, BCM5752, BCM5714, AND BCM5715 ONLY)

Spare Control 1

This version of the register is only applicable to the BCM5705, BCM5721, BCM5751 and BCM5752. The following is enabled by register 1Ch with the shadow value in bits [14:10] = 00010.

Table 670: Spare Control 1 Register (Address 1Ch, Shadow Value 00010)

Bit	Field	Description	Init	Access
15	Write Enable	• 1 = Write bits [9:0]. • 0 = Read bits [9:0].	0	R/W
14:10	Shadow Register Selector	00010 = Spare Control 1 register.	00010	R/W
9:1	Reserved	Write as 00h, ignore when read.	00h	R/W
0	Link LED Mode	• 1 = Enable link LED mode. • $\overline{\text{LINKSPD[2:1]}}$ = speed. • 00: 1000BASE-T link. • 01: 100BASE-TX link. • 10: 10BASE-T link or no link. • $\overline{\text{SLAVE}}$ = Active low 10/100/1000BASE-T link. • 0 = Normal link mode.	0	R/W

Write Enable

During a write to this register, setting Spare Control 1 register bit 15 to a 1 allows writing to bits [7:0] 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 00010 to enable read/write to the Spare Control 1 register address 1Ch.

Link LED Mode

Bit 0 of MII register 1Ch with shadow value 00010 selects the link LED mode. When this bit is set, it enables the link LED mode. The $\overline{\text{LINKSPD2}}$/$\overline{\text{LINKSPD1}}$ are Link/Speed LED and $\overline{\text{SLAVE}}$ LED is $\overline{\text{LINK}}$ LED to indicate a link for 10BASE-T, 100BASE-TX or 1000BASE-T. When this bit is cleared, the $\overline{\text{LINKSPD2}}$, $\overline{\text{LINKSPD1}}$, and $\overline{\text{SLAVE}}$ are in their normal mode.

This version of the register is applicable only to the BCM5714 and BCM5715.

Table 671: Spare Control 1 Register (Shadow Register Selector = 00010)

Bit	Field	Description	Init	Access
15	Write Enable	• 1 = Write bits [9:0]. • 0 = Read bits [9:0].	0	R/W
14:10	Shadow Register Selector	Shadow Register Selector	00010	R/W
9:7	Reserved	Write as 00h, ignore when read.	00	R/W
6	CFC_INITFILTER	1 = enable cfc_initFilter signal to control clock gating of 1000t clocks. Do not gate off 1000t clocks whenever cfiltercntl is initializing the filter.	0	R/W
5	Reserved	write as 0, ignore on read	0	R/W
4	100Base-FX Mode Copper Path	• 1 = enable 100Base-FX on TRD± pins • 0 = normal copper operation on mdi pairs	0	R/W
3	XMT CRC Enable	• 1 = transmit CRC checker enabled • 0 = transmit CRC checker disabled	0	R/W
2	Bicolor Link Speed LED Mode	• 1 = enable Bicolor Link Speed LED mode LINKSPD[1:0] = speed • 10 = 1000BASE-T • 01 = 100BASE-T • 11 = auto-negotiation, 10BASE-T	0	R/W
1	Lost Token Fix Enable	When 0, enables lost token fix reset circuits	0	R/W
0	Link LED Mode	• 1 = enable Link LED mode: • LINKSPD[1:0] = speed • 00 = 1000BASE-T • 01 = 100BASE-T • 10 = 10BASE-T • 11 = auto-negotiation • SLAVE = active low link • 0 = normal link/slave mode	0	R/W

Clock Alignment Control

This version of the register is only applicable to the BCM5705, BCM5721, BCM5751 and BCM5752. The following is enabled by register 1Ch with shadow value in bits [14:10] = 00011.

Table 672: Clock Alignment Control Register (Address 1Ch, Shadow Value 00011)

Bit	Field	Description	Init	Access
15	Write Enable	• 1 = Write bits [9:0]. • 0 = Read bits [9:0].	0	R/W
14:10	Shadow Register Selector	00011 = Clock Alignment Control register.	00011	R/W
9	GTXCLK Clock Delay Enable	• 1 = Enable GTXCLK delay. • 0 = Normal mode (bypass GTXCLK delay).	GTXCLKDLY pin	R/W
8:0	Reserved	Write as 000h, ignore when read.	000h	R/W

Write Enable

During a write to this register, setting Clock Alignment register bit 15 to a 1 allows writing to bits [7:0] 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 register 1Ch.

GTXCLK Clock Delay Enable

Setting bit 9 of MII register 1Ch with shadow value 00011 enables the GTXCLK internal delay. When this bit is cleared, the GTXCLK delay is bypassed.

This version of the register is applicable only to BCM5714 and BCM5715.

Table 673: Clock Alignment Control Register (Shadow Register Selector = 00011)

Bit	*Field*	*Description*	*Init*	*Access*
15	Write Enable	• 1 = Write bits [9:0]. • 0 = Read bits [9:0].	0	R/W
14:10	Shadow Register Selector	Shadow Register Selector	00011	R/W
9	GTXCLK Delay Bypass Disable	• 0 = bypass gtxclk delay • 1 = do not bypass gtxclk delay	0	R/W
8	GMII Clock Alignment Strobe	Delay value is latched into selected GMII clock delay line on rising edge of this bit	0	R/W
7	RXCLK Alignment Strobe	Delay value is latched into selected RX clock delay line on rising edge of this bit	0	R/W
6:4	Delay Value	RXCLK delay: reset = default delay • 000 = +1 unit delay ... • 111 = +8 units delay • GMII clock delay: • 110 = -1.0ns • 111 = -0.5ns • 000 = 0ns • 001 = 0.5ns • 010 = 1.0ns	000	R/W
3:0	Delay Line Selector	RXCLK strobe: • xx00 = std cell rxclk • xx01 = dfse rxclk • xx10 = dfe rxclk • xx11 = enc rxclk • GMII clock strobe: • 0111 = TBI gtx_clk • 1000 = GMII gtx_clk • 1001 = RGMII gtx_clk • 1010 = GMII rx_clk • 1011 = RGMII rx_clk • 1100 = TBI RBC0 • 1101 = TBI RBC1	0000	R/W

Spare Control 2

This version of the register is only applicable to the BCM5705, BCM5721, BCM5751 and BCM5752. The following is enabled by register 1Ch with shadow value in bits [14:10] = 00100.

Table 674: Spare Control 2 Register (Address 1Ch, Shadow Value 00100)

Bit	Field	Description	Init	Access
15	Write Enable	• 1 = Write bits [9:0]. • 0 = Read bits [9:0].	0	R/W
14:10	Shadow Register Selector	00100 = Spare Control 2 register.	00100	R/W
9:5	Reserved	Write as 00h, ignore when read.	00h	R/O
4:2	Reserved	Write as 011, ignore when read.	011	R/O
1	Energy Detect on INTR pin	• 1 = routes Energy Detect to interrupt signal. Use LED selectors (reg 1Ch shadow 01101 and 01110) and program to INTR mode. • 0 = INTR pin is Interrupt function.	0	R/W
0	Reserved	Write as 0, ignore when read.	0	R/O

Write Enable

During a write to this register, setting 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 Spare Control 2 register.

Energy Detect on INTR Pin

Setting bit one of this register enables the Energy Detect function on the INTR pin. Otherwise, the INTR pin defaults to Interrupt function.

This version of the register is applicable only to BCM5714 and BCM5715.

Table 675: Spare Control 2 Register (Shadow Register Selector = 00100)

Bit	Field	Description	Init	Access
15	Write Enable	• 1 = Write bits [9:0]. • 0 = Read bits [9:0].	0	R/W
14:10	Shadow Register Selector	Shadow Register Selector	00100	R/W
9:8	Reserved	write as 0, ignore on read	-	R/W
7	Test on Bits 23:16	• 1 = enable bits 23:16 of test output bus	0	R/W
6	Disable PHY A2	• 1 = internally disable phya2 input (consult testability document for suggested usage)	0	R/W
5	enable rbc0/1 & txc/rxc tristate	• 1 = enable tristating of rbc0/1 or txc/rxc • 0 = rbc0/1 & txc/rxc not tristated	0	R/W
4:2	Wirespeed Retry Limit	• 000: downgrade after 2 failed link attempts • 001: downgrade after 3 failed link attempts ... • 111: downgrade after 9 failed link attempts	011	R/W
1	Energy Detect On INTR Pin	• 1 = routes Energy Detect to interrupt signal. Use LED selectors (reg 1c shadow 01101 and 01110) to direct interrupt signal to an LED output.	0	R/W
0	Test on Bits 7:0	• 1 = enable low byte of test output bus (consult testability document for suggested usage) (aka. testonbyte7_0)	0	R/W

Spare Control 3

This version of the register is only applicable to the BCM5705, BCM5721, BCM5751 and BCM5752. The following is enabled by register 1Ch with shadow value in bits [14:10] = 00101.

Table 676: Spare Control 3 Register (Address 1Ch, Shadow Value 00101)

Bit	Field	Description	Init	Access
15	Write Enable	• 1 = Write bits [9:0]. • 0 = Read bits [9:0].	0	R/W
14:10	Shadow Register Selector	00101 = Spare Control 3 Register.	00101	R/W
9:2	Reserved	Write as 03h, ignore when read.	03h	R/W
1	CLK125 Auto Power-Down	• 1 = Auto power-down of CLK125 is disabled. • 0 = Auto power-down of CLK125 is enabled.	1	R/W
0	CLK125 Output	• 1 = Enable CLK125 output. • 0 = Disable CLK125 output.	1	R/W

Write Enable

During a write to this register, setting 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 Spare Control 3 register.

CLK125 Auto Power Down

Clearing this bit enables the auto power down of the CLK125 output. 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.

This version of the register is applicable only to BCM5714 and BCM5715.

1

Table 677: 1Ch: Spare Control 3 Register (Shadow Register Selector = 00100)

Bit	*Field*	*Description*	*Init*	*Access*
15	Write Enable	• 1 = Write bits [9:0]. • 0 = Read bits [9:0].	0	R/W
14:10	Shadow Register Selector	Shadow Register Selector	00101	R/W
9	DLL lock enable during auto-power down	1 = allow time for dll to lock before enabling clocks & analog components. Only applicable when dll is powered down during auto-power down (r1c.5 bit 1 is LOW).	0	R/W
8	txc/rxc disable during auto-power down	• 1 = disable txc/rxc during auto-power down when there's no energy on the cable	0	R/W
7	10Base-T Carrier Reject Filter Enable	• 1 = enable 10BT 15MHz Carrier Rejection Filter	0	R/W
6	TXC Off Enable	• 1 = gates off TXC output in 1000-Base T mode	0	R/W
5	SD/energy detect change mux select	• 1 = interrupt based on energy detection (top level debounced energy detect change or filtered fiber signal detect change via en_10B pin) • 0 = normal ipphone interrupt selected	0	R/W
4	Low Power ENC Disable	• 1 = disable low power ENC mode	1	R/W
3	Disable Low Power 10Base-T Link Mode	• 1 = disable low power 10Base-T link mode	1	R/W
2	SIGDET Deassert Timer Lengthen	• 1 = 100TX Sigdet Deassert Timer = 40 us • 0 = Sigdet Deassert Timer = 0.25 us	1	R/W
1	Auto-Power Down DLL Off Disable	• 1 = disable powering down of the dll during auto-power down • 0 = enable powering down of dll during auto-power down	1	R/W
0	CLK125 Output Enable	• 1 = enable CLK125 output • 0 = disable CLK125 output	1	R/W

LED Status

This register is only applicable to the BCM5705, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715. The following is enabled by register 1Ch with shadow value in bits [14:10] = 01000.

Table 678: LED Status Register (Address 1Ch, Shadow Value 01000)

Bit	Field	Description	Init	Access
15	Write Enable	• 1 = Write bits [9:0]. • 0 = Read bits [9:0].	0	R/W
14:10	Shadow Register Selector	01000 = LED Status register.	01000	R/W
9	Reserved	Write as 0, ignore when read.	0	R/W
8	$\overline{\text{Slave}}$ Indicator	• 1 = Master mode. • 0 = Slave mode.	0	R/O
7	$\overline{\text{FDX}}$ Indicator	• 1 = Half-duplex mode. • 0 = Full-duplex mode.	0	R/O
6	$\overline{\text{INTR}}$ Indicator	• 1 = No active Interrupt. • 0 = Interrupt activated.	0	R/O
5	Reserved	Write as 0, ignore when read.	0	R/O
4:3	$\overline{\text{LINKSPD}}$ Indicator	• 11 = No Link established. • 10 = 10BASE-T Link established. • 01 = 100BASE-TX Link established. • 00 = 1000BASE-T Link established.	00	R/O
2	$\overline{\text{Transmit}}$ Indicator	• 1 = No transmit activity. • 0 = Transmit activity.	0	R/O
1	$\overline{\text{Receive}}$ Indicator	• 1 = Not receive activity. • 0 = Receive activity.	0	R/O
0	$\overline{\text{Quality}}$ Indicator	• 1 = Quality is not good. • 0 = Quality is good.	0	R/O

Write Enable

During a write to this register, setting LED Status register bit 15 to a 1 allows writing to bits [7:0] 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 LED Status register.

$\overline{\text{Slave}}$ Indicator

When LED Status register 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.

$\overline{\text{FDX}}$ Indicator

When LED Status register 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.

$\overline{\text{INTR}}$ Indicator

When LED Status register 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.

$\overline{\text{LINKSPD}}$ Indicator

When LED Status register 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.

$\overline{\text{Transmit}}$ Indicator

When LED Status register 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.

$\overline{\text{Receive}}$ Indicator

When LED Status register 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.

$\overline{\text{Quality}}$ Indicator

When LED Status register 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.

LED Control

This register is only applicable to the BCM5705, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715. The following is enabled by register 1Ch with shadow value in bits [14:10] = 01001.

Table 679: LED Control Register (Address 1Ch, Shadow Value 01001)

Bit	Field	Description	Init	Access
15	Write Enable	• 1 = Write bits [9:0]. • 0 = Read bits [9:0].	0	R/W
14:10	Shadow Register Selector	01001 = LED Control register.	01001	R/W
9:6	Reserved	Write as 00000, ignore when read.	00000	R/W
5	Override GBIC LED mode (BCM5714 and BCM5715 only)	• 1 = LEDs not remapped in GBIC mode. • 0 = In GBIC mode LEDs mapped as follows: - LED1: RX_LOSS - LED2: RX - LED3: TX - LED4: LINK	0	R/W
	Reserved (Other Devices)	Write as 0, ignore when read	0	R/O
4	Activity/Link LED Enable	• 1 = Drive activity/link data on $\overline{\text{ACTIVITY}}$ LED. • 0 = Drive activity data on $\overline{\text{ACTIVITY}}$ LED. **Note:** This bit overrides bit 3 below.	0	R/W
3	$\overline{\text{ACTIVITY}}$ LED Enable	• 1 = Drive activity data on $\overline{\text{ACTIVITY}}$ LED. • 0 = Drive receive data on $\overline{\text{ACTIVITY}}$ LED.	1	R/W
2	Remote Fault LED Enable	• 1 = Drive remote fault on quality LED. • 0 = Normal operation.	0	R/W
1:0	Link Utilization LED Selector	• 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. **Note:** This mode has higher priority than the activity LED enable mode in bit 3.	00	R/W

Write Enable

During a write to this register, setting 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

LED Control register bits [14:10] of this register must be set to 01001 to enable read/write to the LED Control register 1Ch.

Activity/Link LED Enable

Setting LED Control register bit 4 drives activity/link data on $\overline{\text{ACTIVITY}}$ LED.

$\overline{\text{ACTIVITY}}$ LED Enable

Setting LED Control register bit 3 drives activity data on $\overline{\text{ACTIVITY}}$ LED. Otherwise, it drives receive data on $\overline{\text{ACTIVITY}}$ LED.

Remote Fault LED Enable

Setting LED Control register bit 2 drives remote fault on quality LED.

Link Utilization LED Selector

These bits apply to the LED programmed to the ACTIVITY mode only. In the 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 to10%, 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

Auto Power-Down

This register is only applicable to the BCM5705, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715. The following is enabled by register 1Ch with shadow value in bits [14:10] = 01010.

Table 680: Auto Power-Down Register (Address 1Ch, Shadow Value 01010)

Bit	Field	Description	Init	Access
15	Write Enable	• 1 = Write bits [9:0]. • 0 = Read bits [9:0].	0	R/W
14:10	Shadow Register Selector	01010 = Auto Power Down register.	01010	R/W
9:6	Reserved	Write as 0h, ignore when read.	0h	R/W
5	Auto Power Down Mode	• 1 = Auto power down mode enabled. • 0 = Auto power down mode disabled.	0	R/W
4	Sleep Timer Select	• 1 = Sleep timer is 5.4s. • 0 = Sleep timer is 2.7s.	0	R/W
3:0	Wakeup Timer Select	Counter for Wakeup timer in units of 84 ms. • 0001 = 84 ms. • 0010 = 168 ms. • ... • 1111 = 1.26s.	0001	R/W

Write Enable

During a write to this register, setting 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 Auto Power Down register address 1Ch.

Auto Power-Down Mode Enable

Setting this bit enables the Auto Power-Down Mode.

Sleep Timer Select

Setting this bit changes the wakeup time leaving Auto Power-Down Mode.

Wakeup Timer Select

The port continues wakeup mode for a time based on the count stored in this register. The minimum value is 84 ms and the maximum value is 1.26s. This only applies when the part is in Auto Power-Down mode.

LED Selector 1

This register is only applicable to the BCM5705, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715. The following is enabled by register 1Ch with shadow value in bits [14:10] = 01101.

Table 681: LED Selector 1 Register (Address 1Ch, Shadow Value 01101)

Bit	*Field*	*Description*	*Init*	*Access*
15	Write Enable	• 1 = Write bits [9:0]. • 0 = Read bits [9:0].	0	R/W
14:10	Shadow Register Selector	01101 = LED Selector 1 register.	01101	R/W
9:8	Reserved	Write as 00, ignore when read.	00	R/W
7:4	LED2 Selector	• 0000 = $\overline{\text{LINKSPD[1]}}$ • 0001 = $\overline{\text{LINKSPD[2]}}$ • 0010 = $\overline{\text{XMITLED}}$ • 0011 = $\overline{\text{ACTIVITY}}$ • 0100 = $\overline{\text{FDXLED}}$ • 0101 = $\overline{\text{SLAVE}}$ • 0110 = $\overline{\text{INTR}}$ • 0111 = $\overline{\text{QUALITY}}$ • 1000 = $\overline{\text{RCVLED}}$ • 1001 = Reserved • 1010 = $\overline{\text{MULTICOLOR[2]}}$ • 1011 = $\overline{\text{OPENSHORT}}$ • 1100 = $\overline{\text{ENERGYLNK}}$ • 1101 and 1110 = Off (high) • 1111 = On (low)	0001	R/W
3:0	LED1 Selector	• 0000 = $\overline{\text{LINKSPD[1]}}$ • 0001 = $\overline{\text{LINKSPD[2]}}$ • 0010 = $\overline{\text{XMITLED}}$ • 0011 = $\overline{\text{ACTIVITY}}$ • 0100 = $\overline{\text{FDXLED}}$ • 0101 = $\overline{\text{SLAVE}}$ • 0110 = $\overline{\text{INTR}}$ • 0111 = $\overline{\text{QUALITY}}$ • 1000 = $\overline{\text{RCVLED}}$ • 1001 = Reserved • 1010 = $\overline{\text{MULTICOLOR[1]}}$ • 1011 = $\overline{\text{OPENSHORT}}$ • 1100 = $\overline{\text{ENERGYLNK}}$ • 1101 and 1110 = Off (high) • 1111 = On (low)	0000	R/W

Write Enable

During a write to this register, setting LED Selector 1 register bit 15 to a 1 allows writing to bits [7:0] 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 LED Selector register 1 address 1Ch.

LED2 Selector

Bits [7:4] of MII register 1Ch with shadow value 01101 select the LED2 output mode.

LED1 Selector

Bits [3:0] of MII register 1Ch with shadow value 01101 select the LED1 output mod.

LED Selector 2

This register is only applicable to the BCM5705, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715. The following is enabled by register 1Ch with shadow value in bits [14:10] = 01110.

Table 682: LED Selector 2 Register (Address 1Ch, Shadow Value 01110)

Bit	Field	Description	Init	Access
15	Write Enable	• 1 = Write bits [9:0]. • 0 = Read bits [9:0].	0	R/W
14:10	Shadow Register Selector	01110 = LED Selector 2 register.	01110	R/W
9:8	Reserved	Write as 00, ignore when read.	00	R/W
7:4	LED4 Selector	• 0000 = $\overline{\text{LINKSPD[1]}}$ • 0001 = $\overline{\text{LINKSPD[2]}}$ • 0010 = $\overline{\text{XMITLED}}$ • 0011 = $\overline{\text{ACTIVITY}}$ • 0100 = $\overline{\text{FDXLED}}$ • 0101 = $\overline{\text{SLAVE}}$ • 0110 = $\overline{\text{INTR}}$ • 0111 = $\overline{\text{QUALITY}}$ • 1000 = $\overline{\text{RCVLED}}$ • 1001 = Reserved • 1010 = $\overline{\text{MULTICOLOR[2]}}$ • 1011 = $\overline{\text{OPENSHORT}}$ • 1100 = $\overline{\text{ENERGYLNK}}$ • 1101 and 1110 = Off (high) • 1111 = On (low)	0110	R/W

Table 682: LED Selector 2 Register (Address 1Ch, Shadow Value 01110) (Cont.)

Bit	Field	Description	Init	Access
3:0	LED3 Selector	• 0000 = $\overline{\text{LINKSPD[1]}}$ • 0001 = $\overline{\text{LINKSPD[2]}}$ • 0010 = $\overline{\text{XMITLED}}$ • 0011 = $\overline{\text{ACTIVITY}}$ • 0100 = $\overline{\text{FDXLED}}$ • 0101 = $\overline{\text{SLAVE}}$ • 0110 = $\overline{\text{INTR}}$ • 0111 = $\overline{\text{QUALITY}}$ • 1000 = $\overline{\text{RCVLED}}$ • 1001 = Reserved • 1010 = $\overline{\text{MULTICOLOR[1]}}$ • 1011 = $\overline{\text{OPENSHORT}}$ • 1100 = $\overline{\text{ENERGYLNK}}$ • 1101 and 1110 = Off (high) • 1111 = On (low)	0011	R/W

Write Enable

During a write to this register, setting LED Selector 2 register bit 15 to a 1 allows writing to bits [7:0] 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 LED Selector register 2 address 1Ch.

LED4 Selector

Bits [7:4] of MII register 1Ch with shadow value 01110 select the LED2 output mode.

LED3 Selector

Bits [3:0] of MII register 1Ch with shadow value 01110 select the LED1 output mode.

LED GPIO Control/Status

This register is only applicable to the BCM5705, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715. The following is enabled by register 1Ch with shadow value in bits [14:10] = 01111.

Table 683: LED GPIO Control/Status Register (Address 1Ch, Shadow Value 01111)

Bit	Field	Description	Init	Access
15	Write Enable	• 1 = Write bits [9:0]. • 0 = Read bits [9:0].	0	R/W
14:10	Shadow Register Selector	01111 = LED GPIO Control/Status register.	01111	R/W
9:8	Reserved	Write as 00, ignore when read.	00	R/W
7:4	LED I/O Status	• Bit 7 = LED4 pin status. • Bit 6 = LED3 pin status. • Bit 5 = LED2 pin status • Bit 4 = LED1 pin status. • 1 = LED pin is an input. • 0 = LED pin is an output.	0h	R/O
3:0	Programmable LED I/O Control	• Bit 3 = LED4 pin control. • Bit 2 = LED3 pin control. • Bit 1 = LED2 pin control. • Bit 0 = LED1 pin control. • 1 = Disable LED output enable. • 0 = Enable LED output enable.	0h	R/W

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 LED GPIO Control/Status register 1Ch.

LED I/O Status

LED GPIO Control/Status register bits [7:4] read back the status of the LED pin.

Programmable LED I/O Control

Setting LED GPIO Control/Status register bits [3:0] set the LED pin to disable LED output. Clearing LED GPIO Control/Status register bits [3:0] set the LED pin to enable LED output.

Autodetect SGMII/Media Converter

This register is only applicable to the BCM5705, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715. The following is enabled by register 1Ch with shadow value in bits [14:10] = 11000.

Table 684: Autodetect SGMII/Media Converter Register (Address 1Ch, Shadow Value 11000)

Bit	Field	Description	Init	Access
15	Write Enable	• 1 = Write bits [9:0]. • 0 = Read bits [9:0].	0	R/W
14:10	Shadow Register Selector	11000 = Autodetect SGMII/Media Converter register.	11000	R/W
9	SerDes Resolution Fault	• 1 = Selected field mismatch. • 0 = No mismatch or SGMII/media converter autodetect mode is disabled.	0	R/O
8:3	Reserved	Write as 00h, ignore when read.	00h	R/O
2	1000T PCS transmit FIFO Elasticity (BCM5714 and BCM5715 only)	• 1 = Support jumbo packets • 0 = low elasticity (low latency)	1	R/W
	Reserved (Other Devices)	Write as 0, ignore when read	0	R/O
1	SGMII 10/100 RX FIFO Frequency Lock Mode (BCM5714 and BCM5715 only)	• 1 = SGMII Rx FIFO will assume that the SerDes-recovered clock and the local clock are frequency locked. This will essentially bypass the fifo with the lowest possible latency in 10/100 speeds (useful for applications where the MAC/switch and PHY are using the same crystal) • 0 = normal operation	0	R/W
	Reserved (Other Devices)	Write as 0, ignore when read	0	R/O
0	SGMII/Media Converter Autodetect Mode Enable	• 1 = Enable SGMII/media converter autodetect mode. • 0 = Normal operation.	INTF_SEL[3] AND RXCDLY	R/W

Write Enable

During a write to this register, setting Autodetect SGMII/Media Converter 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 11000 to enable read/write to the Autodetect SGMII/Media Converter address 1Ch.

SerDes Resolution Fault

Bit 9 of the Auto-Detect SGMII/Media Converter register indicates there is a selected field mismatch on bit 0 of the base page word. Otherwise, it reads a 0.

SGMII/Media Converter Autodetect Mode Enable

Setting Bit 0 of the Auto-Detect SGMII/Media Converter register enable the SGMII/media converter autodetect mode. Otherwise, it is in the normal mode.

1000BASE-X Auto-Negotiation Debug

This register is only applicable to the BCM5705, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715. The following is enabled by register 1Ch with shadow value in bits [14:10] = 11010.

Table 685: 1000BASE-X Auto-Negotiation Debug Register (Address 1Ch, Shadow Value 11010)

Bit	Field	Description	Init	Access
15	Write Enable	• 1 = Write bits [9:0]. • 0 = Read bits [9:0].	0	R/W
14:10	Shadow Register Selector	11010 = 1000BASE-X Auto-negotiation Debug register.	11010	R/W
9	Consistency Mismatch	• 1 = Consistency mismatch occurred since last read. • 0 = No consistency mismatch occurred since last read.	0	R/O LH
8	RUDI Invalid	• 1 = RUDI invalid detected since last read. • 0 = No RUDI invalid detected since last read.	0	R/O LH
7	Comma Detected	• 1 = Comma detected since last read. • 0 = No comma detected since last read.	0	R/O LH
6	AN_Sync_Status	• 1 = AN_sync_status has not failed since last read. • 0 = AN_sync_status failed since last read.	0	R/O LH
5	Idle Detect State	• 1 = Idle detect state entered since last read. • 0 = Idle detect state has not been entered since last read.	0	R/O LH
4	Complete Acknowledge State	• 1 = Complete acknowledge state entered since last read. • 0 = Complete acknowledge state has not been entered since last read.	0	R/O LH
3	Acknowledge Detect State	• 1 = Acknowledge detect state entered since last read. • 0 = Acknowledge detect state has not been entered since last read.	0	R/O LH
2	Ability Detect State	• 1 = Ability detect state entered since last read. • 0 = Ability detect state has not been entered since last read.	0	R/O LH
1	Error State	• 1 = Error state entered since last read. • 0 = Error state has not been entered since last read.	0	R/O LH
0	AN_Enable State	• 1 = AN_enable state entered since last read. • 0 = AN_enable state has not been entered since last read.	0	R/O LH

Write Enable

During a write to this register, setting LED 1000BASE-X Auto-negotiation Debug register bit 15 to a 1 allows writing to bits [7:0] 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 of 1000BASE-X Auto-negotiation Debug register indicates a consistency mismatch occurred since last read.

RUDI Invalid

Bit 8 of 1000BASE-X Auto-negotiation Debug register indicates a RUDI (Rx_UnitData Indicate, specified in IEEE 36.2.5.1.6) invalid detected since last read.

Comma Detected

Bit 7 of 1000BASE-X Auto-negotiation Debug register indicates a comma was detected since last read.

AN_Sync_Status

Bit 6 of 1000BASE-X Auto-negotiation Debug register indicates the AN_sync_status has not failed since last read.

Idle Detect State

Bit 5 of 1000BASE-X Auto-negotiation Debug register indicates the idle detect state entered since last read.

Complete Acknowledge State

Bit 4 of 1000BASE-X Auto-negotiation Debug register indicates the complete acknowledge state entered since last read.

Acknowledge Detect State

Bit 3 of 1000BASE-X Auto-negotiation Debug register indicates the acknowledge detect state entered since last read.

Ability Detect State

Bit 2 of 1000BASE-X Auto-negotiation Debug register indicates the ability detect state entered since last read.

Error State

Bit 1 of 1000BASE-X Auto-negotiation Debug register indicates the error state entered since last read.

AN_Enable State

Bit 0 of 1000BASE-X Auto-negotiation Debug register indicates the AN_enable state entered since last read.

Auxiliary 1000BASE-X Control

This register is only applicable to the BCM5705, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715. The following is enabled by register 1Ch with shadow value in bits [14:10] = 11011.

Table 686: Auxiliary 1000BASE-X Control Register (Address 1Ch, Shadow Value 11011)

Bit	Field	Description	Init	Access
15	Write Enable	• 1 = Write bits [9:0]. • 0 = Read bits [9:0].	0	R/W
14:10	Shadow Register Selector	11011 = Auxiliary 1000BASE-X Control register.	11011	R/W
9	Use SerDes Mode Counters	• 1 = Use registers 12–14h for SerDes data. • 0 = Normal operation.	0	R/W
8	AutoNeg Fast Timers (BCM5714 and BCM5715 only)	• 1 = speed up link_timer for test vectors (1.6 us SGMII; 9.84 us IEEE) • 0 = Normal operation	0	R/W
	Reserved (Other Devices)	Write as 0, ignore when read	0	R/O
7	Jam False Carrier Mode (BCM5714 and BCM5715 only)	• 1 = Send packet with txen, txer, txd=55h for duration of false carrier in SGMII/GBIC half-duplex mode • 0 = ignore false carriers in SGMII/GBIC mode	1	R/W
	Reserved (Other Devices)	Write as 0, ignore when read	0	R/O
6	Disable Carrier Extend (BCM5714 and BCM5715 only)	1 = force rxer, rxd to zeros in TRR+extend state (pcs receive state)	0	R/W
	Reserved (Other Devices)	Write as 0, ignore when read	0	R/O
5	Disable TRRR (BCM5714 and BCM5715 only)	• 1 = bypass extend_by_1 state (pcs transmit state) • 0 = normal operation	0	R/W
	Reserved (Other Devices)	Write as 0, ignore when read	0	R/O
4	Disable Remote Fault Sensing	• 1 = Disable automatic remote fault sensing of auto-negotiation resolution error. • 0 = Normal Operation.	0	R/W
3	AutoNeg Error Timer Enable (BCM5714 and BCM5715 only)	• 1 = enable autoneg error timer (error state entered when error timer expires in ability_detect, acknowledge_detect, or idle_detect state) • 0 = normal operation	0	R/W
	Reserved (Other Devices)	Write as 0, ignore when read	0	R/O
2	Comma Detect Enable	• 1 = Enable comma detection. • 0 = Disable comma detection.	1	R/W
1	FIFO Elasticity	• 1 = High elasticity to support jumbo packets (supports 10/100/1000 jumbo packets). • 0 = Low elasticity (low latency).	1	R/W
0	Disable CRC Checker	• 1 = Disable CRC checker. • 0 = Enable CRC checker.	1	R/W

Write Enable

During a write to this register, setting Auxiliary 1000BASE-X Control register 2 bit 15 to a 1 allows writing to bits [7:0] 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 SerDes Mode Counters

Setting bit 9 of the Auxiliary 1000BASE-X Control register enables SerDes data to be presented on register 12h–14h.

Disable Remote Fault Sensing

Setting bit 4 of the Auxiliary 1000BASE-X Control register disables automatic remote fault sensing of an auto-negotiation resolution error.

Comma Detect Enable

Setting bit 2 of Auxiliary 1000BASE-X Control register enables comma detection.

FIFO Elasticity

Setting bit 1 of Auxiliary 1000BASE-X Control register enables 10/100/1000 Mbps jumbo packet reception while in SGMII mode. In this mode, the BCM57XX can transmit packets up to 9 KB in length. When this bit is cleared, the FIFO elasticity is set to low latency. In this mode, the BCM57XX can transmit packets up to 4.5 KB in length.

Disable CRC Checker

Setting bit 0 of Auxiliary 1000BASE-X Control register disables the CRC checker.

Auxiliary 1000BASE-X Status

This register is only applicable to the BCM5705, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715. The following is enabled by register 1Ch with shadow value in bits [14:10] = 11100.

Table 687: Auxiliary 1000BASE-X Status Register (Address 1Ch, Shadow Value 11100)

Bit	Field	Description	Init	Access
15	Write Enable	• 1 = Write bits [9:0]. • 0 = Read bits [9:0].	0	R/W
14:10	Shadow Register Selector	11100 = Auxiliary 1000BASE-X Status register.	11100	R/W
9	Link Status Change	• 1 = Link status change has occurred since last read. • 0 = Link status change has not occurred since last read.	0	R/O LH
8	SGMII Selector Mismatch	• 1 = SGMII selector mismatch in SGMII mode. • 0 = SerDes, copper, GBIC mode, or SGMII selector does not mismatch, or auto-negotiation is disabled.	0	R/O
7	Auto-negotiation Resolution Error	• 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	R/O
6:5	Link Partner Remote Fault	• Reflects 1000BASE-X register 05h [13:12]. • 00 = No remote fault. • 10 = Off line. • 01 = Link fault. • 11 = Auto-negotiation error.	00	R/O
4	Auto-negotiation Page Received	• 1 = Page has been received since last read. • 0 = Page has not been received since last read.	0	R/O LH
3	Current Operating Duplex Mode	• 1 = PHY is operating in full-duplex mode. • 0 = PHY is operating in half-duplex mode (or auto-negotiation has not completed).	0	R/O
2	Link Status	• 1 = Link is up on SerDes side. • 0 = Link is down on SerDes side.	0	R/O
1	PAUSE Resolution—Receive Side	• 1 = Enable pause receive. • 0 = Disable pause receive.	0	R/O
0	PAUSE Resolution—Transmit Side	• 1 = Enable pause transmit. • 0 = Disable pause transmit.	0	R/O

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 of Auxiliary 1000BASE-X Status register indicates that the link status has changed since the last register read.

SGMII Selector Mismatch

Bit 8 of Auxiliary 1000BASE-X Status register indicates an SGMII selector mismatch in SGMII mode.

Auto-Negotiation Resolution Error

Bit 7 of Auxiliary 1000BASE-X Status register indicates auto-negotiation HCD is none (no common half-duplex or full-duplex abilities).

Link Partner Remote Fault

Bits 6 and 5 of Auxiliary 1000BASE-X Status register indicates the link partner remote fault status reflected from 1000BASE-X register 05h bits 13:12.

Auto-Negotiation Page Received

Bit 4 of Auxiliary 1000BASE-X Status register indicates auto-negotiation page has been received since last read.

Current Operating Duplex Mode

Bit 3 of Auxiliary 1000BASE-X Status register indicates the PHY is operating in full-duplex mode.

Link Status

Bit 2 of Auxiliary 1000BASE-X Status register indicates the PHY link is up on the SerDes side.

PAUSE Resolution—Receive Side

Bit 1 of Auxiliary 1000BASE-X Status register indicates receive pause resolution.

PAUSE Resolution—Transmit Side

Bit 0 of Auxiliary 1000BASE-X Status register indicates transmit pause resolution.

Miscellaneous 1000BASE-X Status

This register is only applicable to the BCM5705, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715. The following is enabled by register 1Ch with shadow value in bits [14:10] = 11101.

Table 688: Miscellaneous 1000BASE-X Status Register (Address 1Ch, Shadow Value 11101)

Bit	Field	Description	Init	Access
15	Write Enable	• 1 = Write bits [9:0]. • 0 = Read bits [9:0].	0	R/W
14:10	Shadow Register Selector	11101 = Misc 1000BASE-X Status register.	11101	R/W
9	TX FIFO Error	• 1 = Transmit FIFO error since last read. • 0 = No transmit FIFO error since last read.	0	R/O LH
8	RX FIFO Error	• 1 = Receive FIFO error since last read. • 0 = No receive FIFO error since last read.	0	R/O LH
7	Bad FIFO Pointer (BCM5714 and BCM5715 only)	• 1 = fifo pointer all zeros since last read • 0 = bad fifo pointer has not occurred since	0	R/O LH
	Reserved (Other Devices)	Write as 0, ignore when read	0	R/O
6	False Carrier Jammed (BCM5714 and BCM5715 only)	• 1 = false carrier jammed in SGMII/GBIC mode since last read • 0 = no false carrier jammed or mode is disabled via register 1ch shadow 27 [7]	0	R/O LH
	Reserved (Other Devices)	Write as 0, ignore when read	0	R/O
5	False Carrier Detected	• 1 = False carrier detected since last read. • 0 = No false carriers detected since last read.	0	R/O LH
4	CRC Error Detected	• 1 = CRC Error detected since last read. • 0 = No CRC error detected since last read or mode is disabled via register 1Ch, shadow 11011, bit 0.	0	R/O LH
3	Transmit Error Detected	• 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	R/O LH
2	Receive Error Detected	• 1 = Receive error since last read (early_end state in PCS receive). • 0 = No receive error since last read.	0	R/O LH
1	Carrier Extend Error Detected	• 1 = Carrier extend error since last read (extend_err state in PCS receive). • 0 = No carrier extend error since last read.	0	R/O LH
0	Early End Extension Detected	• 1 = Early end extension since last read (early_end_ext state in PCS receive). • 0 = No early end extension since last read.	0	R/O LH

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 of Misc 1000BASE-X Status register indicates a transmit FIFO error since the last read.

RX FIFO Error

Bit 8 of Misc 1000BASE-X Status register indicates a receive FIFO error since the last read.

False Carrier Detected

Bit 5 of Misc 1000BASE-X Status register indicates a false carrier detected since the last read.

CRC Error Detected

Bit 4 of Misc 1000BASE-X Status register indicates a CRC error detected since the last read.

Transmit Error Detected

Bit 3 of Misc 1000BASE-X Status register indicates a transmit error code detected since the last read.

Receive Error Detected

Bit 2 of Misc 1000BASE-X Status register indicates a receive error code detected since the last read.

Carrier Extend Error Detected

Bit 1 of Misc 1000BASE-X Status register indicates a carrier extend error since the last read.

Early End Extension Detected

Bit 0 of Misc 1000BASE-X Status register indicates an early end extension since the last read.

Autodetect Medium

This register is only applicable to the BCM5705, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715. The following is enabled by register 1Ch with shadow value in bits [14:10] = 11110.

Table 689: Autodetect Medium Register (Address 1Ch, Shadow Value 11110)

Bit	Field	Description	Init	Access
15	Write Enable	• 1 = Write bits [9:0]. • 0 = Read bits [9:0].	0	R/W
14:10	Shadow Register Selector	11110 = Autodetect Medium register.	11110	R/W
9	Reserved	Write as 0, ignore when read	0	R/O
8	Invert Signal Detect From Pin (BCM5714 and BCM5715 only)	1 = invert SerDes signal detect from pin 0 = normal operation (active high pin)	0	R/W
	Reserved (Other Devices)	Write as 0, ignore when read	0	R/O
7	SerDes In-Use LED Mode	• 1 = Drive transmit LED active low when SerDes is selected, inactive when copper selected. • 0 = Normal transmit LED operation.	NOT INTF_SEL[3]	R/W
6	SerDes LED mode	• 1 = Use SerDes transmit, receive, and link for LEDs whenever SerDes mode is selected via register 1Ch, shadow 11111, bits [2:1]. • 0 = Always use copper transmit, receive, and link for LEDs regardless of the mode selected.	INTF_SEL[2] AND NOT INTF_SEL[3]	R/W
5	Qualify SerDes Signal Detect (BCM5714 and BCM5715 only)	• 1 = SerDes signal detect from pin is ANDed with sync status • 0 = normal operation	0	R/W
	Reserved (Other Devices)	Write as 1, ignore when read	0	R/O
4	SerDes Auto Power Down Mode	1 = Power down SerDes when filtered signal detect is inactive. 0 = Normal operation.	0	R/W
3	Power Down Inactive Interface	1 = Power down SerDes when copper is selected; power down copper when SerDes is selected. 0 = Normal operation.	0	R/W
2	Autodetect Media Default	1 = SerDes selected when no medium is active. 0 = Copper selected when no medium is active.	0	R/W
1	Autodetect Medium Priority	1 = SerDes selected when both media are active. 0 = Copper selected when both media are active.	1	R/W
0	Autodetect Medium Enable	1 = Enable autodetect medium. 0 = Disable autodetect medium.	0	R/W

Write Enable

During a write to this register, setting the 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 read/write to the Autodetect Medium register.

SerDes In-Use LED Mode

Bit 7 of the Autodetect Medium register drives the transmit LED active low when SerDes is selected; inactive when copper is selected.

SerDes LED Mode

Bit 6 of the Autodetect Medium register enables the use SerDes transmit, receive, and link for LEDs whenever SerDes mode is selected via register 1Ch, shadow 11111, bits [2:1].

SerDes Auto Power Down mode

Bit 4 of the Autodetect Medium register enables the power down of SerDes when the filtered signal detect is inactive.

Power Down Inactive Interface

Bit 3 of the Autodetect Medium register enables the power down of SerDes when copper is selected, and power down of copper when SerDes is selected.

Autodetect Media Default

Bit 2 of the Autodetect Medium register enables the selection of SerDes as default medium when no medium is active. Clearing this bit sets copper as the default medium.

Autodetect Media Priority

Bit 1 of the Autodetect Medium register enables the selection of SerDes priority when both media are active. Clearing this bit sets copper as the default medium selection.

Autodetect Media Enable

Bit 0 of the Autodetect Medium register enables the autodetect media function.

Mode Control

This register is only applicable to the BCM5705, BCM5721, BCM5751, BCM5752, BCM5714, and BCM5715. The following is enabled by register 1Ch with shadow value in bits [14:10] = 11111.

Table 690: Mode Control Register (Address 1Ch, Shadow Value 11111)

Bit	Field	Description	Init	Access
15	Write Enable	• 1 = Write bits [9:0]. • 0 = Read bits [9:0].	0	R/W
14:10	Shadow Register Selector	11111 = Mode Control register.	11111	R/W
9	Reserved	Write as 0, ignore when read.	0	R/O
8	Mode Select Change	• 1 = Interface Mode Select status changed since last read. • 0 = Interface Mode Select status did not change since last read.	0	R/O LH
7	Copper Link	• 1 = Link is good on the copper interface. • 0 = copper link is down.	0	R/O
6	SerDes Link	• 1 = Link is good on the SerDes interface. • 0 = SerDes link is down.	0	R/O
5	Copper Energy Detect	• 1 = Energy detected on the copper interface. • 0 = Energy not detected on the copper interface.	0	R/O
4	Signal Detect (SerDes Mode)	• 1 = Filtered energy detected on the SerDes interface. • 0 = Energy not detected on the SerDes interface.	0	R/O
3	SerDes Capable (BCM5714 and BCM5715 only)	• 1 = SerDes capable device • 0 = not SerDes capable device	0	R/O
	Reserved (Other Devices)	Write as 1, ignore when read	0	R/O
2:1	Mode Select	• 00 = Copper • 01 = SerDes • 10 = SGMII • 11 = Media converter	INTF_SEL[3:2]	R/W
0	Enable 1000BASE-X Registers	• 1 = Select 1000BASE-X registers for addresses 00h–0Fh. • 0 = Select copper registers for addresses 00h–0Fh.	INTF_SEL[3:2] = 01	R/W

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 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 of the Mode Control register indicates that there is change in the interface mode selection. Otherwise, it reads a 0.

Copper Link

Bit 7 of the Mode Control register indicates that the link status of the copper interface is up. Otherwise, it reads a 0.

SerDes Link

Bit 6 of the Mode Control register indicates the link status of the SerDes interface is up. Otherwise, it reads a 0.

Copper Energy Detect

Bit 5 of the Mode Control register indicates that energy is detected in the copper interface. Otherwise, it reads a 0.

Signal Detect (SerDes Mode)

Bit 4 of the Mode Control register indicates that signal energy is detected in the SerDes interface. Otherwise, it reads a 0.

Mode Select

Bits 2:1 of the Mode Control register select one of the four available interfaces.

Enable 1000BASE-X Registers

Setting bit 0 of the Mode Control register 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.

HCD Status Register (PHY_Addr = 0x1, Reg_Addr = 1Dh, Bit 15 = 1)

Table 691: HCD Status Register (PHY_Addr = 0x1, Reg_Addr = 1Dh, Bit 15 = 1)

Bit	Field	Description	Init	Access
15	Enable Shadow Register	• 1 = Select Shadow register. • 0 = Normal operation.	0	R/W
14	Wirespeed Disable Gigabit Advertising (BCM5714 and BCM5715 only)	1 = disable advertising gigabit 0 = advertise gigabit based on register 9	0	R/O
	Reserved (Other Devices)	Write as 0, ignore when read	0	R/O
13	Wirespeed Disable 100TX (BCM5714 and BCM5715 only)	• 1 = disable advertising 100B-TX • 0 = advertise 100B-TX based on register 4	0	R/O
	Reserved (Other Devices)	Write as 0, ignore when read	0	R/O
12	Reserved (BCM5705, BCM5721, and BCM5751 only)			
	Wire Speed downgrade (Other devices)	• 1 = Wire Speed down grade occurred since last read. • 0 = Wire Speed down grade cleared.	0	R/O LH
11	HCD 1000BASE-T FDX	• 1 = Gigabit Full-duplex occurred since last read. • 0 = HCD Cleared.	0	R/O LH
10	HCD 1000BASE-T	• 1 = Gigabit Half-duplex occurred since last read. • 0 = HCD Cleared.	0	R/O
9	HCD 100BASE-T FDX	• 1 = 100BASE-TX Full-duplex occurred since last read. • 0 = HCD Cleared.	0	R/O LH
8	HCD 100BASE-T	• 1 = 100BASE-TX Half-duplex occurred since last read. • 0 = HCD Cleared.	0	R/O LH
7	HCD 10BASE-T FDX	• 1 = 10 BASE-T Full-duplex occurred since last read. • 0 = HCD Cleared.	0	R/O LH
6	HCD 10BASE-T	• 1 = 10 BASE-T Half-duplex occurred since last read. • 0 = HCD Cleared.		
5	HCD 1000BASE-T FDX (Link never came up)	• 1 = Gigabit Full-duplex HCD and Link never came up occurred since last read. • 0 = HCD Cleared.	0	R/O LH
4	HCD 1000BASE-T (Link never came up)	• 1 = Gigabit Half-duplex HCD and Link never came up occurred since last read. • 0 = HCD Cleared.	0	R/O LH
3	HCD 100BASE-T FDX (Link never came up)	• 1 = 100BASE-TX Full-duplex HCD and Link never came up occurred since last read. • 0 = HCD Cleared.	0	R/O LH
2	HCD 100BASE-T (Link never came up)	• 1 = 100BASE-TX Half-duplex HCD and Link never came up occurred since last read. • 0 = HCD Cleared.	0	R/O LH
1	HCD 10BASE-T FDX (Link never came up)	• 1 = 10BASE-T Full-duplex HCD and Link never came up occurred since last read Full-duplex. • 0 = HCD Cleared.	0	R/O LH

Table 691: HCD Status Register (PHY_Addr = 0x1, Reg_Addr = 1Dh, Bit 15 = 1) (Cont.)

Bit	Field	Description	Init	Access
0	HCD 10BASE-T (Link never came up)	• 1 = 10BASE-T Half-duplex HCD and Link never came up occurred since last read. • 0 = HCD Cleared.	0	R/O LH

MASTER/SLAVE SEED REGISTER (PHY_ADDR = 0X1, REG_ADDR = 1DH, BIT 15 = 0)

Table 692: Master/Slave Seed Register (PHY_Addr = 0x1, Reg_Addr = 1Dh, Bit 15 = 0)

Bit	Name	Description	Init	Access
15	Enable Shadow Register	• 1 = Select Shadow register. • 0 = Normal operation.	0	R/W
14	Master/Slave Speed Match	• 1 = Seeds match. • 0 = Seeds don't match.	0	R/O, LH
13	Link partner Repeater/DTE Bit	• 1 = Link partner is a repeater/switch device port. • 0 = Link partner is a DTE device port.	0	R/O
12	Link Partner Manual M/S Config Value	• 1 = Link partner is configured as master. • 0 = Link partner is configured as slave.	0	R/O
11	Link Partner Manual M/S Config Enable	• 1 = Link partner manual master/slave configuration enabled. • 0 = Link partner manual master/slave configuration disabled.	0	R/O
10:0	Local Master/Slave Seed Value	Returns the automatically generated Master/Slave random seed.	000h	R/W

PHY TEST REGISTER 1 (PHY_ADDR = 0X1, REG_ADDR = 1EH)

This version of the register is applicable to BCM5714 and BCM5715 only.

Table 693: PHY Test Register 1 (PHY_Addr = 0x1, Reg_Addr = 1Eh)

Bit	*Name*	*Description*	*Init*	*Access*
15	CRC Error Count Visibility	• 1 = receiver NOT_OK counters merged into one 16 bit counter to count CRC errors instead (crc errors will only be counted after this bit is set) • 0 = normal operation	0	R/W
14	Transmit Error Code Visibility	• 1 = false carrier sense counter counts packets received with transmit error codes instead (errors will only be counted after this bit is set) • 0 = normal operation	0	R/W
13	Counter Test Mode	• 1 = forces counters into test mode • 0 = normal operation	0	R/W
12	Force Link	• 1 = force link state machine into pass state • 0 = normal operation	0	R/W
11	Force Lock	• 1 = force descrambler into locked state • 0 = normal operation	0	R/W
10	Scrambler Test	• 1 = speed up descrambler unlock detect timer • 0 = normal operation	0	R/W
9	External Link	• 1 = use tpin11 input as link status • 0 = normal operation	0	R/W
8	Fast Timers	• 1 = timers are sped up for LSI test • 0 = normal operation	0	R/W
7	Manual Swap MDI State	• 1 = swap • 0 = off	0	R/W
6	Receive Watchdog Timer Disable	• 1 = watchdog timer disabled • 0 = reset receive PMD when descrambler can't lock within 730us of link or lock loss.	0	R/W
5	Disable Polarity Encode	• 1 = disable 1000Base-T polarity encoding • 0 = normal operation	0	R/W
4	Enable Software Trim Setting (Main DAC)	• 1 = use software trim setting • 0 = use hardware trim setting ***Note:*** 1) register setting from slice1 is used to control main dac trim. 2) register setting from slice2 is used to control hybrid dac trim.	0	R/W
3:0	Trim[3:0] (Main DAC)	Software trim setting ***Note:*** only slice1 register setting is used. 2) bit 1 from trim pad is inverted	0	R/W

This version of the register is applicable to rest of the NetXtreme family.

Table 694: PHY Test Register 1 (PHY_Addr = 0x1, Reg_Addr = 1Eh)

Bit	*Name*	*Description*	*Init*	*Access*
15	CRC Error Count Visibility	• 1 = Receiver NOT_OK counters (see "Receiver NOT_OK Counters (PHY_Addr = 0x1, Reg_Addr = 14h)" on page 629) merged into one 16-bit CRC error counter. • 0 = Normal operation.	0	R/W
14:8	Reserved		0	R/W
7	Manual Swap MDI State (BCM5705, BCM5721, and BCM5751 only)	• 1 = Manual swap MDI state. • 0 = Normal operation.	0	R/W
	Reserved		0	R/W
6:0	Reserved	Write as 0, Ignore when read.	0	R/W

PHY Test Register 2 (PHY_ADDR = 0X1, REG_ADDR = 1FH)

This register is applicable to the BCM5714 and BCM5715.

Table 695: 1Fh: Test Register 2

Bit	*Name*	*Description*	*Init*	*Access*
15:13	Test Select Auto-Negotiation FSM	• 000 = ARB • 001 = RX1000 • 010 = RX • 011 = TX1000 • 100 = TX • 101 = BASET LINK	000	R/W
12	Test Auto-Negotiation Timer	• 1 = auto-negotiation timer test mode • 0 = normal operation	0	R/W
11	Test Master/Slave Seed	• 1 = use MDIO programmable master/slave seed • 0 = normal operation	0	R/W
10	Writeable Link Partner Ability	• 1 = link partner advertised ability may be overwritten by MII management • 0 = normal operation	0	R/W
9	Force HCD	• 1 = force auto-negotiation HCD resolution (hcd can only be checked in register 19h bits [10:8]; hcd status register will not be updated) • 0 = normal operation	0	R/W
8	Writeable Link Partner M/S Seed	• 1 = link partner master/slave seed may be overwritten by MII management • 0 = normal operation	0	R/W
7	Transmit 10B Mode	• 1 = force GMII transmit into 10B mode • 0 = use normal mode bit	0	R/W
6	Receive 10B Mode	• 1 = force GMII receive into 10B mode • 0 = use normal mode bit	0	R/W

Table 695: 1Fh: Test Register 2 (Cont.)

Bit	Name	Description	Init	Access
5	Bypass Transmit FIFO	• 1 = transmit FIFO bypassed • 0 = normal operation	0	R/W
4	Same Scrambler Seeds	• 1 = receive scrambler uses transmit seed • 0 = normal operation	0	R/W
3	Jitter Test Mode	• 1 = jitter test mode • 0 = normal operation	0	R/W
2	Test ATMP Counter	• 1 = force master slave seed attempt counter into test mode • 0 = normal mode	0	R/W
1	Latency Measure	• 1 = send special short packet to measure receive latency on remote PHY (hold high) • 0 = normal operation	0	R/W
0	Disable Active Hybrid	• 1 = active hybrid disabled • 0 = active hybrid enabled	0	R/W

Appendix A: Flow Control

NOTES

Developers can refer to the IEEE 802.3 Annex 31B specification for detailed information on Ethernet flow control mechanisms.

- Flow control frames use a well-known multicast address, defined in the 802.1D Bridging specification The MAC destination address is 01-80-C2-00-00-01
- Bridges and Switches will not forward pause frames to downstream ports.
- A pause frame contains a request_operand that contains a pause_time field. Pause_time specifies the number of quanta, which transmission should be inhibited.
- Pause frames cannot inhibit MAC control Frames
- Pause_time is a two-octet field, which represents a quanta value. The quanta value is based on bit/slot times for the connection speed. Valid pause_times vary from 0 to 65535.
- The pause frame contains a MAC control opcode. 00-01 is reserved for PAUSE MAC control functions.
- MAC control layers will provide two indicators—paused and not paused.
- The Enet source address equals the unicast address of the MAC sublayer, which transmits the pause_frame.
- The receive engine will set a countdown timer, based on the value of pause_time. When the timer expires, the transmit engine may resume send operation
- A Mac sublayer may transmit pause frames with pause_time = 0. The zero value will stop a pause count down, executed by the MAC's link partner. Effectively, a value of zero restarts a link partner's transmit engine, assuming the link partner was inhibited by a previous pause operation.

FLOW CONTROL SCENARIO

This scenario assumes that the Gigabit switch has a 1:1 port mapping, between the Gigabit Server and Client. The switch does not implement header aligned blocking, nor will it drop packets to alleviate buffer pressure. The following constraints are placed on this scenario:

- Client
  - Full-duplex connection at Gigabit speed.
  - Implements flow control.
  - Flow control enabled.
- Switch
  - Does not drop packets.
  - Full-duplex connection to Client.
- Server
  - Gigabit connection.
  - Either half/full-duplex connection at Gigabit speed (i.e., this scenario will cover two subcases).

File Transfer

The client begins a FTP session (see Figure 109). The file size is very large and will take several minutes for a complete transfer, even at gigabit wire speed.

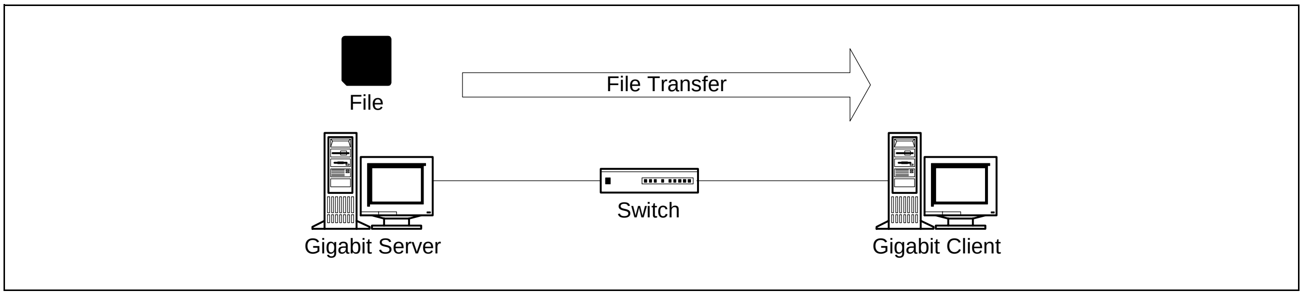


Figure 109: File Transfer Scenario: FTP Session Begins

Speed Mismatch

The Client sends pause frame(s) to the switch (see Figure 110). The Client's pipe has been saturated, and the RX buffers are almost exhausted. The Client begins sending pause frames, when the RX buffer high-water mark/threshold is hit. Any number of reasons can account for the RX buffer issue. The assumption will be made that the Client PCI bus lack bandwidth to DMA packets, at wire speed, to host memory. The user may be playing a DVD, for example.

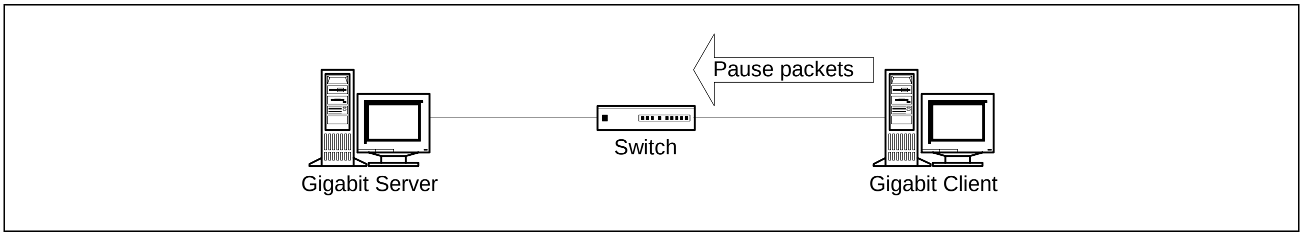


Figure 110: File Transfer Scenario: Speed Mismatch

SWITCH BUFFERS RUN LOW

The switch must wait/inhibit transmission to the Client (see Figure 111). During the pause interval, the Server is still sending packets to the Switch. The Switch will buffer some packets, but will eventually hit an internal threshold; memory will run short. Since dropping packets is undesirable, the Switch must slow incoming packets from the Server. The duplex mode of the Server's connection will dictate how the switch slows traffic. There are two options:

- Jamming (half-duplex)
- Pause frames (full-duplex)

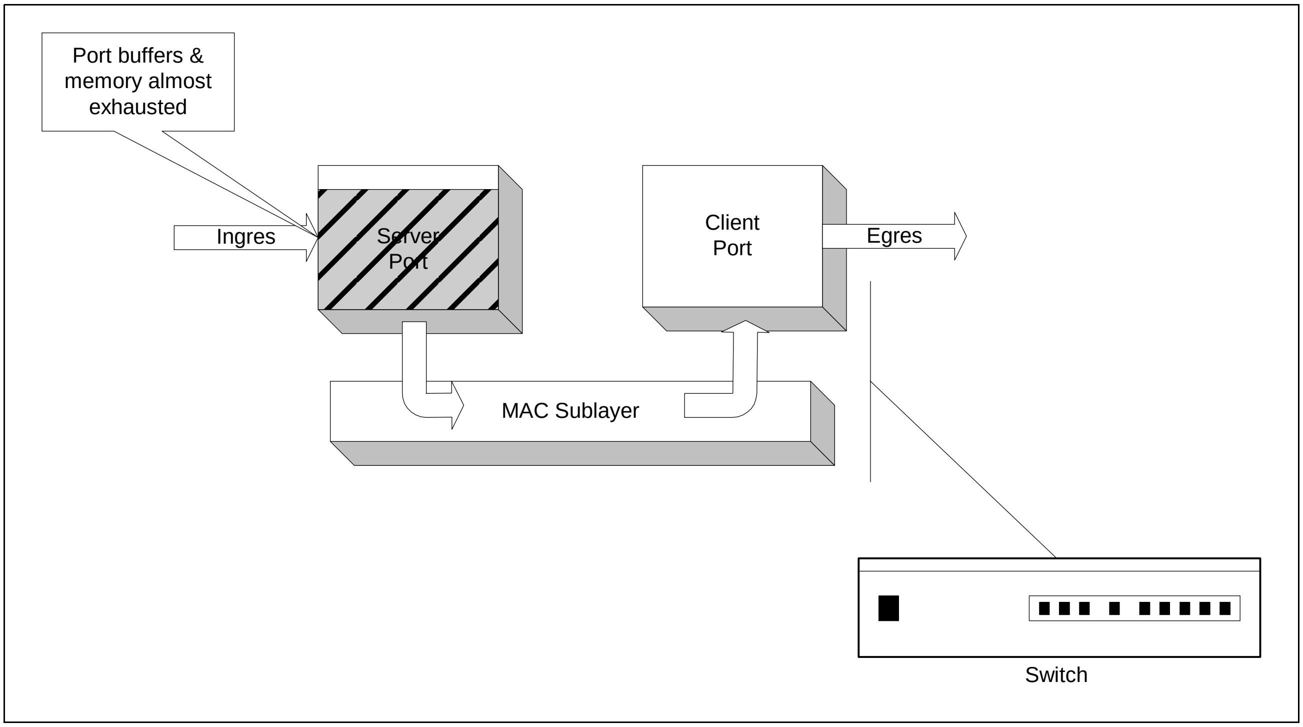


Figure 111: File Transfer Scenario: Speed Buffers Run Low

SWITCH BACKPRESSURE

The Switch will jam ports configured with half-duplex link to slow frame transmission (see Figure 112). In this case, the Server connection must be half-duplex, and then the switch may apply backpressure to the port. The Switch will transmit a jamming pattern, which will prevent the Server from transmitting further packets. The Server’s MAC will detect a collision situation, and will back off for a specified interval. The Switch will continue to apply backpressure to the Server Ingress, until the Client egress is available. The Client port will be available when the pause interval expires, and no further pause packets are sent by the Gigabit Client.

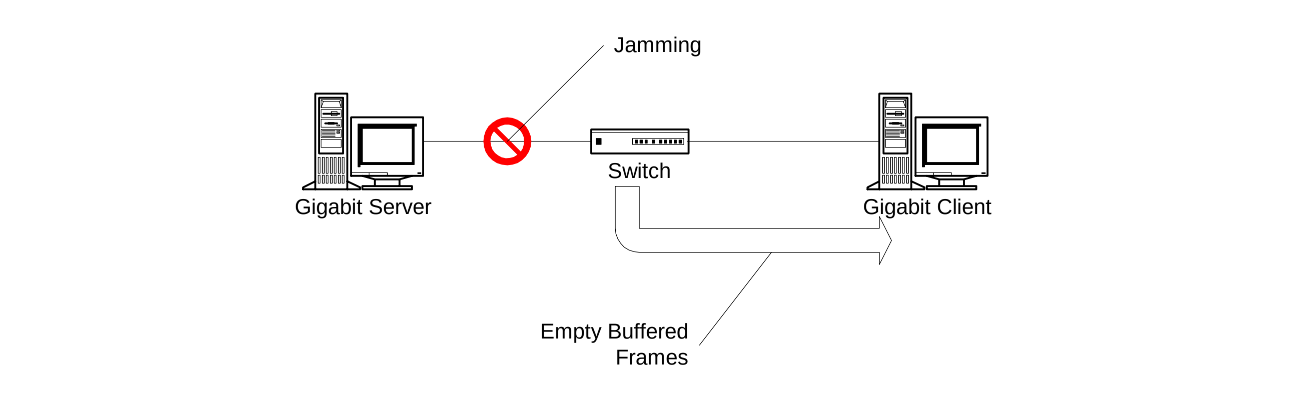


Figure 112: File Transfer Scenario: Switch Backpressure

SWITCH FLOW CONTROL

The Switch can only use IEEE 802.3x flow control when the link is configured for full-duplex operation (see Figure 113). When buffers are near exhaustion, the switch will send a pause frame to the Server. The Server’s MAC will be inhibited for a pause_time interval. During the pause interval, the Switch has the opportunity to empty the buffered packets. Once the buffered packets fall below the high water mark, the Switch may send another pause frame, with pause_time = 0, to terminate the Server’s pause interval. The Switch may also allow the Server’s pause interval to expire. Either way, the Switch no longer will inhibit the Server from sending packets.

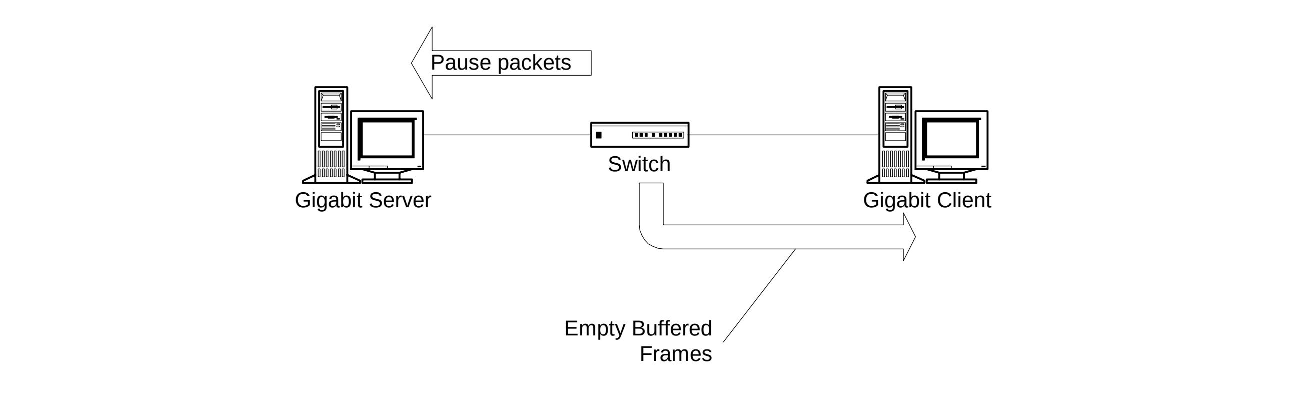


Figure 113: File Transfer Scenario: Switch Flow Control

FILE TRANSFER COMPLETE

The Client has caught up with the transmission flow of the Server (see Figure 114). The Client's RX buffers/memory is below the flow control threshold. The file is transfer is complete. This scenario was a worst-case cascade, where the pause delay propagated through the LAN. The Switch could absorb the Client's pause delay, without having to flow control the Server.

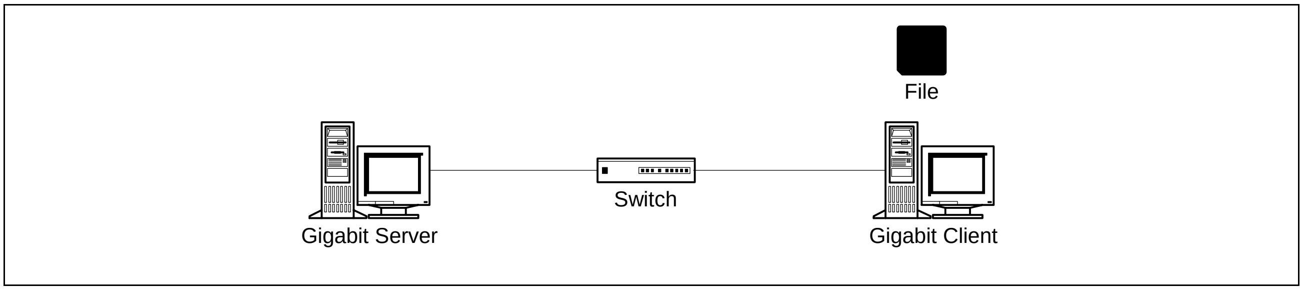
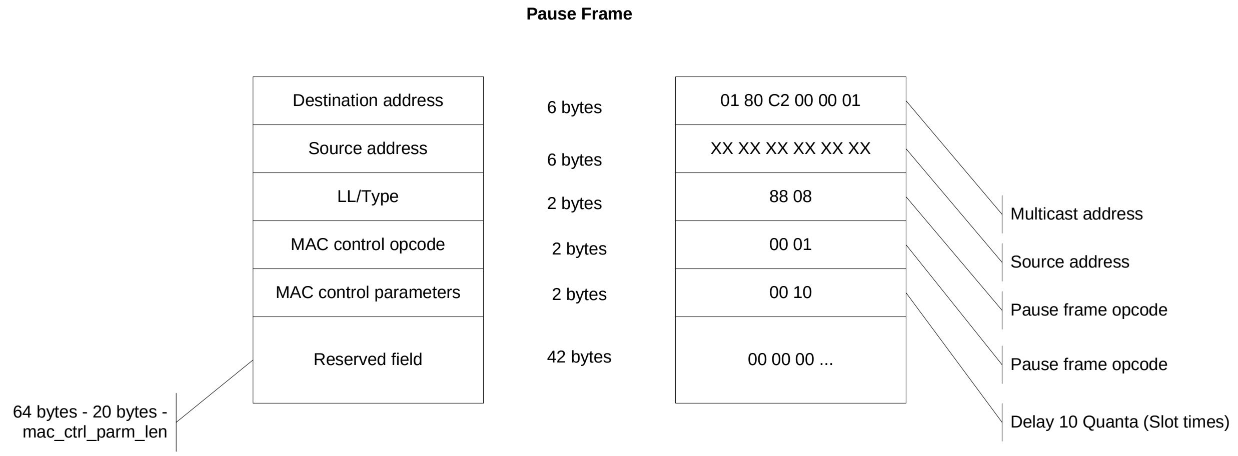


Figure 114: File Transfer Scenario: File Transfer Complete

PAUSE CONTROL FRAME

The minimum size frame is 512 bits or 64 bytes (see Figure 115). MAC control frames must pad zeros into the unused portion of the payload. A flow control frame contains the following fields:

- Destination address field, set to 01-80-C2-00-00-01
- Source address field set to unique MAC address of sender
- LL/Type field set to the 802_3_MAC_CONTROL value, set to 88-08
- MAC control pause opcode (00-01), pause_time, and reserved field (zeros)



Pause Frame

Destination address	6 bytes	01 80 C2 00 00 01
Source address	6 bytes	XX XX XX XX XX XX
LL/Type	2 bytes	88 08
MAC control opcode	2 bytes	00 01
MAC control parameters	2 bytes	00 10
Reserved field	42 bytes	00 00 00 ...

Figure 115: Pause Control Frame

Appendix B: PC Power Management

Reference Materials

The ACPI specification is an industry collaboration between Compaq, Intel®, Microsoft®, Toshiba®, and Phoenix. The following hyperlinks are websites dedicated to the ACPI specification:

- http://developer.intel.com/technology/iapc/acpi/downloads/ACPICA-ProgRef.pdf
- http://www.microsoft.com/hwdev/onnow/
- http://www.toshiba.ca/ISG/newsletter/powermanagement.html
- http://www.phoenix.com/platform/acpi.html
- http://www.teleport.com/~acpi/

The APM specification was authored by Intel and Microsoft. The http://www.microsoft.com/HWDEV/busbios/amp_12.htm website hyperlink(s) contain more information on APM.

Information regarding PCI power management can be located at the PCI Special Interest Group's (SIG) website at http://www.pcisig.com/.

APM

The original Advanced Power Management specification was written in 1992, by Intel and Microsoft. APM is primarily a BIOS interface and has been outdated by the newer ACPI specification. New servers, workstations, and PCs do not implement power management using APM.

APM is a four-layer architecture (see Figure 116):

- Layer 1 is the APM BIOS. The system BIOS provides logic for power management of the motherboard. The BIOS is platform specific software. The APM BIOS may be configured to provide some power management, independent of OS control. Setup and configuration modes for BIOS normally provide a mechanism to disable independent power management by BIOS.
- Layer 2 is the APM interface. A common interface, which uses Int15, calls to the APM BIOS. Additional, 16- and 32-bit protected mode BIOS calls may be published by the BIOS vendor.
- Layer 3 is the OS APM driver. APM aware applications and device drivers interface with the OS APM driver, not the BIOS directly. The APM driver is responsible for serializing access to the APM BIOS, via the APM interface. The OS APM driver must call the APM BIOS once per second so the BIOS will not assume a system hang—watchdog type functionality.
- Layer 4 is the APM aware Device Drivers and applications. NIC vendors like Broadcom provide power management capable device drivers. Add-in devices, like the BCM57XX family, will have a power management capable device driver.

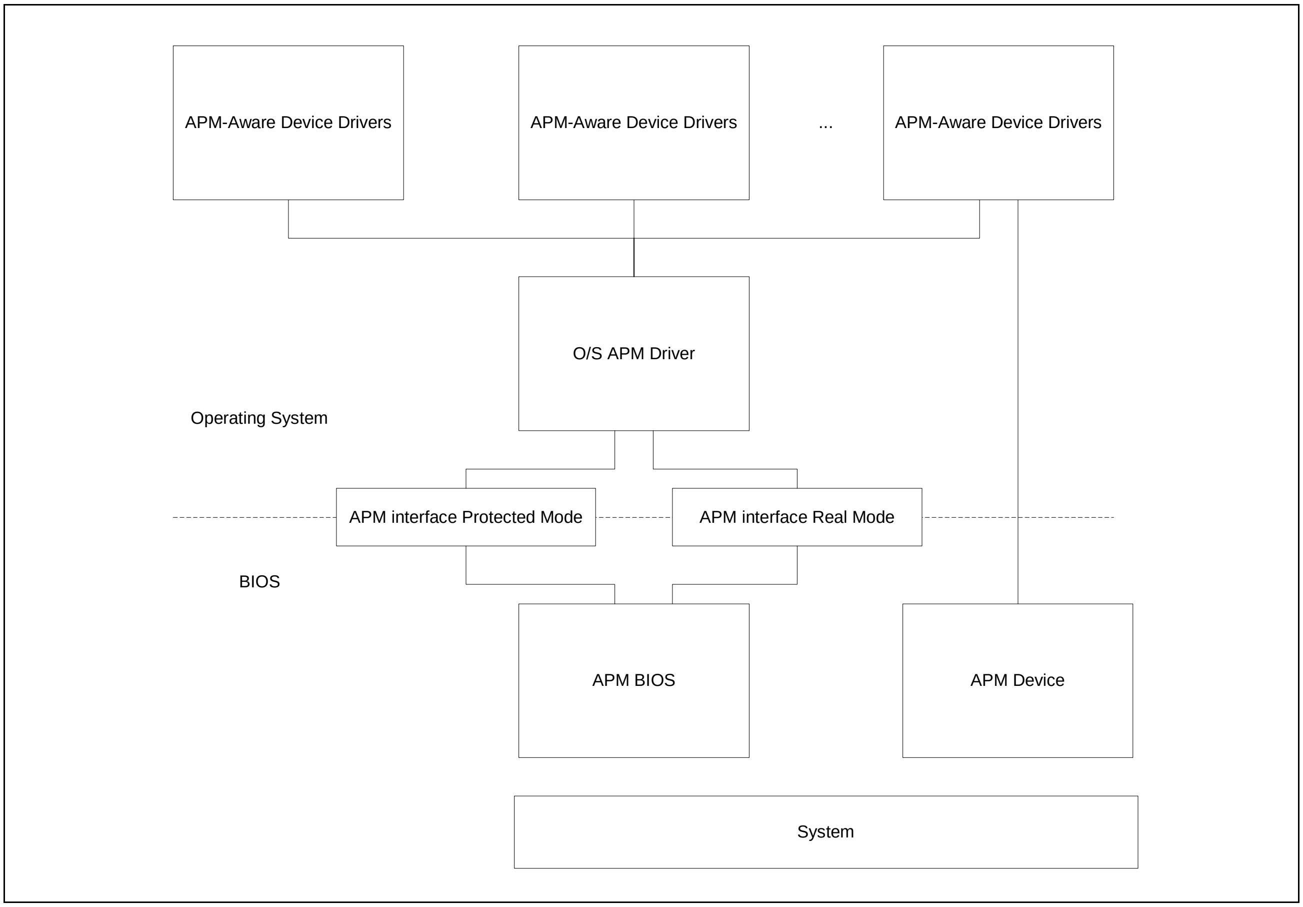


Figure 116: APM Architecture

APM manages system power consumption using six managed states (see Figure 117):

- Full On—System is running at full power. Devices do not implement power conservation measures.
- APM Enabled—System is running under full power. The system does implement power management.
- APM Standby—System running at reduced power. Power managed devices reduce their power consumption to low-power. The processor and/or system clock may be slowed.
- APM Suspend—System is at maximum power conservation. The microprocessor stops clock. CPU core is in low-power mode. Power managed devices are turned off.
- Hibernation—System is completely switched off. Before shutdown, memory is stored to non-volatile storage (i.e., hard drive). Memory modules are turned off.
- Off—No power to system.

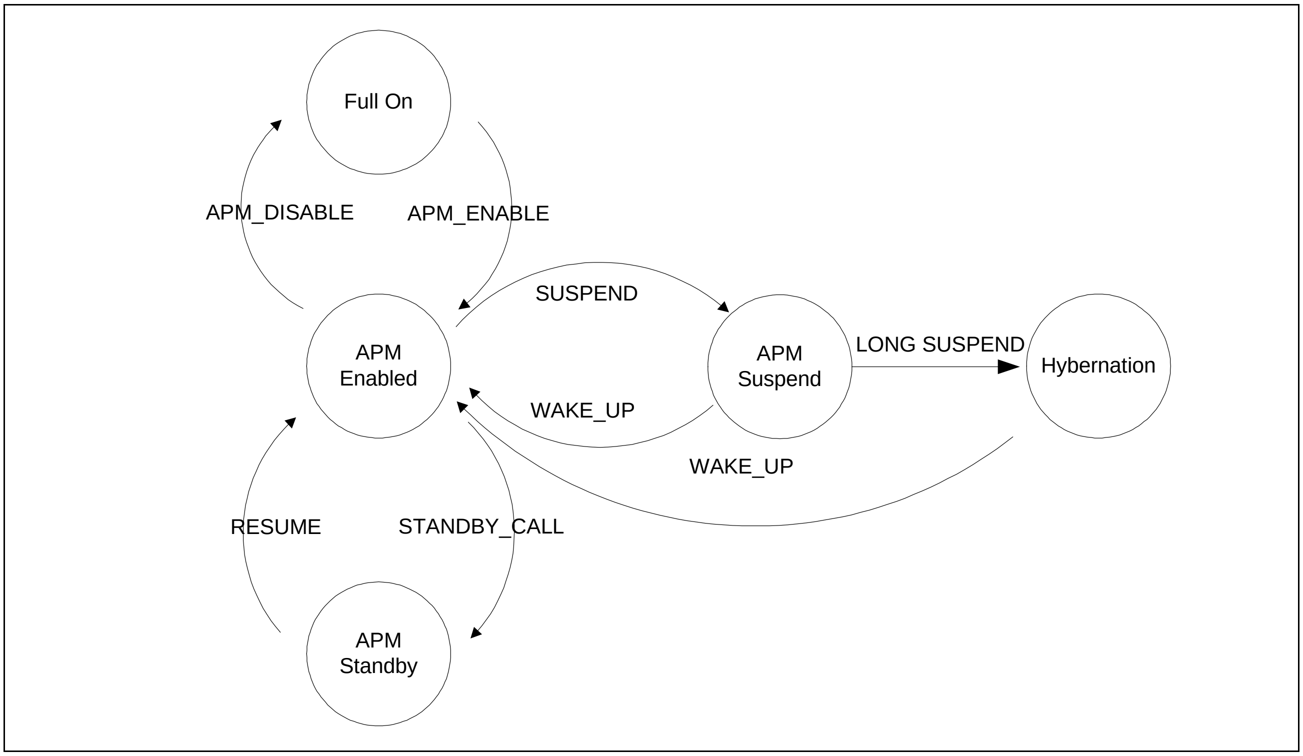


Figure 117: States for Power Consumption Management

The following register interfaces were taken from the APM BIOS specification and are intended as a quick reference (see Table 696). In real mode, these functions are accessed with an Int15h. In protected mode, these functions are called via a protected mode entry point. The entry point is returned from a protected mode connection function call.

To set up a real mode function call, the following steps are taken. Note that these steps assume an x86 host CPU architecture.

- Write signature 0x53 to the AH register
- Write the function code to AL register
- Write the device number to BX register
- Write 0x01(enable) or 0x00(disable) to CX register

Table 696: Function Codes Quick Reference

AH Value	Function Description
0x01	APM Installation Check
0x02	APM real mode interface connect
0x03	APM protected mode connect 16-bit
0x04	APM protected mode connect 32-bit
0x05	APM interface disconnect
0x06	CPU Idle
0x07	CPU busy
0x08	Set power state
0x09	Enable/disable power management
0x0A	Get power status
0x0B	Get power managed event
0x0C	Get power state
0x0D	Enable/disable device power management
0x0E	APM driver version
0x0F	Engage/disengage power management
0x10	Get capabilities
0x11	Get/set/disable/resume timer
0x12	Enable/disable resume on ring indicator
0x13	Enable/disable timer based requests
0x80	OEM APM function

The O/S APM driver must poll the APM BIOS to determine an event code. The O/S driver uses its interface connection to call function Get_power_managed_event (AH = 0x0B). The APM BIOS will acknowledge the call with an event code in register BX. When multiple events occur, the APM BIOS will return all events in a time-ordered sequence. Table 697 is a quick reference for event codes.

Table 697: Event Codes Quick Reference

BX Value	Event Description
0x01	System standby request
0x02	System suspend request
0x03	Normal resume system
0x04	Critical resume system
0x05	Battery low
0x06	Power status change
0x07	Update time
0x08	Critical system suspend
0x09	User system standby request
0x0A	User system suspend request
0x0B	System standby resume
0x0C	Capabilities change
0x00D–0x0FF	Reserved system events
0x100–0x1FF	Reserved device events
0x200–0x2FF	OEM-Defined APM events

Advanced Configuration and Power Interface

The Advanced Configuration and Power Interface (ACPI) specification has displaced APM as the power management standard. APM implements power management in BIOS methods; whereas, ACPI moves power management under the OS umbrella. ACPI is an interface specification between hardware and software. ACPI outlines mechanisms that both hardware and software must follow to remain compatible. The ACPI 2.0 specification was written by Intel, Microsoft, Toshiba, Compaq, and Phoenix Technologies in July 2000.

ACPI can be broken into the following high-level components (see Figure 118):

- OS power management model—The OSPM is responsible for making global decisions regarding power management, of the system.
- Kernel—The OS Kernel provides the basic Device Driver Interfaces (DDI) and runtime code for objects like synchronization, resource sharing, and scheduling.
- Power Managed Device Driver—Vendor supplied software, which programs device-specific power management features. For example, the BCM57XX family needs to disable RX/TX RISC processor clocking for D3 states. The O/S does not have the necessary runtime code or understanding of the MAC's architecture.
- ACPI Driver—A O/S subsystem responsible for ACPI power management. The ACPI driver is responsible for interpreting ACPI Machine Language (AML). AML is typically provided by OEMs for value adds on their platform.
- ACPI Tables—The ACPI tables are descriptions of hardware resources. The interfaces (i.e., registers, memory map), which ACPI system software may use, are exposed through Definition blocks contained within the ACPI tables. The ACPI tables also contain AML event handlers for ACPI system events.
- ACPI Registers—The interfaces described in the ACPI Tables.
- ACPI BIOS—The ACPI BIOS contains the ACPI tables, which describe the hardware hierarchy and resources. The BIOS contains firmware, which boots the PC. Sleep, wake, reset, and event interfaces are contained within the ACPI BIOS.

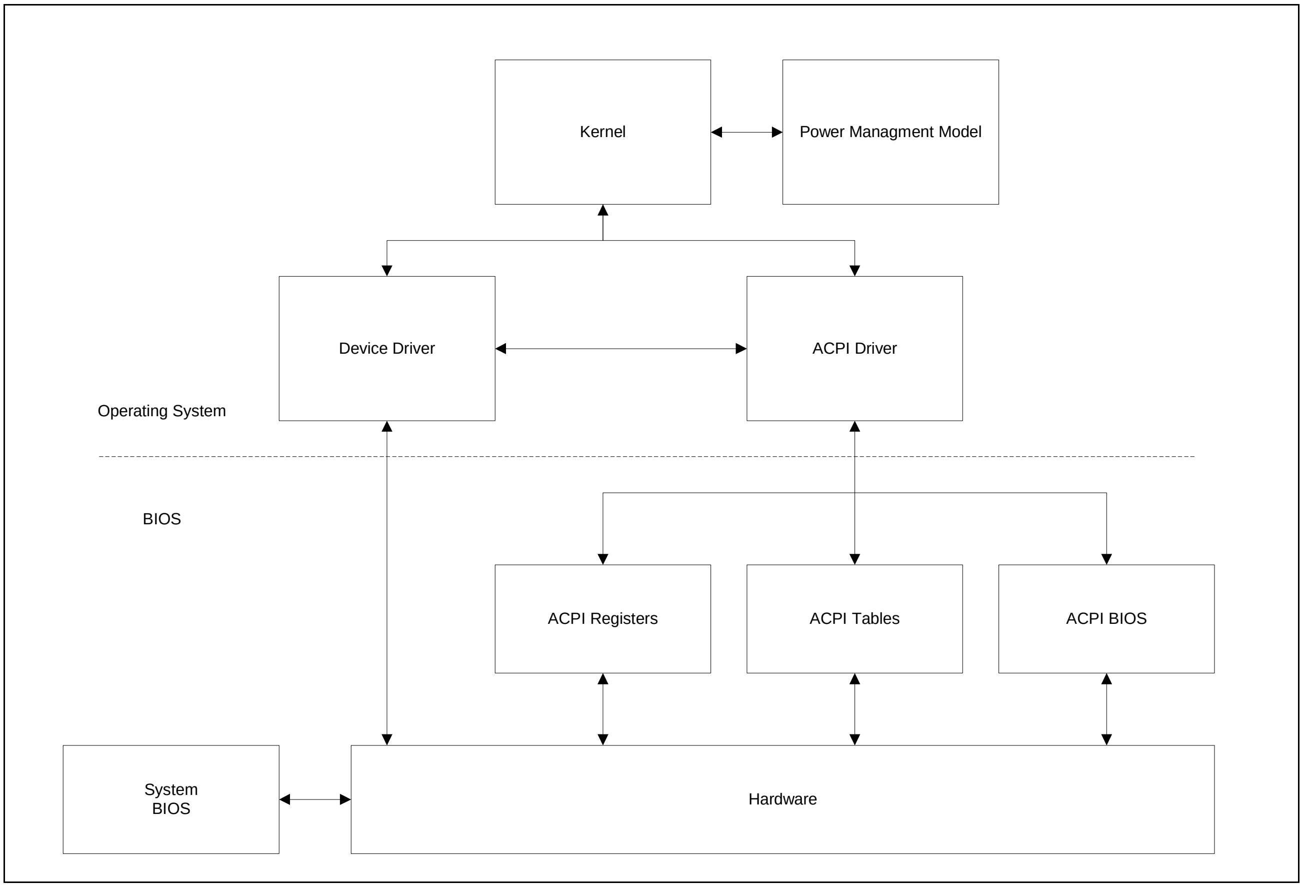


Figure 118: Advanced Configuration and Power Interface (ACPI) Components

The OS Power Management (OSPM) is responsible for global power state transitions. A global state is a platform wide configuration, which directs sleep and device power management state(s). The following global states are defined in the ACPI 2.0 specification (see Figure 119):

- G0 Working—The ACPI 2.0 specification defines the G0 state as follows:

  "The normal operating environment of an ACPI machine. In this state, different devices are dynamically transitioning between their respective power states (D0, D1, D2, D3)."

  Essentially, the OSPM may throttle unused portions of the platform architecture to conserver power. The GO state is synonymous with the Sleep State S0.
- G1 Sleeping—The G1state contains four substates (S1, S2, S3, S4). These substates are platform implementations of different sleep modes. The depth and amount of power savings depends upon the sleep state (S1-S4), which the platform implements.
- G2 Soft Off—The ACPI 2.0 specification defines the G2 state as follows:

  "A computer state where the computer consumes a minimal amount of power."

  System context will not be preserved by hardware.
- G3 Mechanical Off—No power to system. It is safe to replace hardware. There should be zero power consumption.

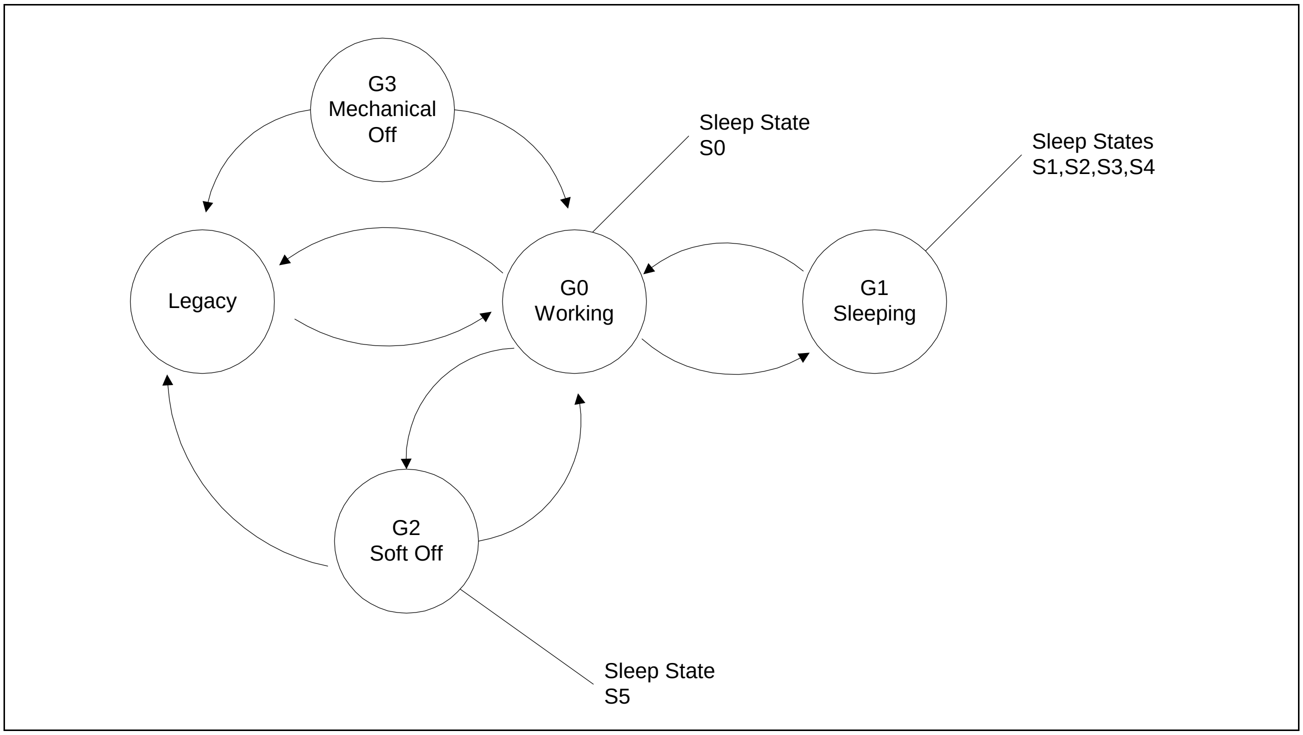


Figure 119: OS Power Management (OSPM) Global States

The ACPI Sleep States are subsets of the global ACPI G0-G3 states. For example, G0 working state corresponds to the S0 sleep state. The sleep states are defined in the ACPI 2.0 specification (see Figure 120):

- S0—normal working state. Devices may be in any D0-D3 state. Hot removal of devices possible.
- S1—low wake-latency sleep state. System context is preserved in this sleep state.
- S2 —low wake-latency sleep state. Memory context is preserved. All remaining system device context may be lost. Wake event capability by devices.
- S3—low wake-latency sleep state. Similar to S2 state. Memory in low-power/refresh mode. Only devices supporting memory are powered.
- S4—lowest power and longest wake-latency sleep state. All devices powered off. Platform context saved off, before entry to S4 state.
- S5—soft off state. This is not a sleep state. No context is preserved, nor can be restored by OSPM. BIOS does a power on reset when a wake event is detected.

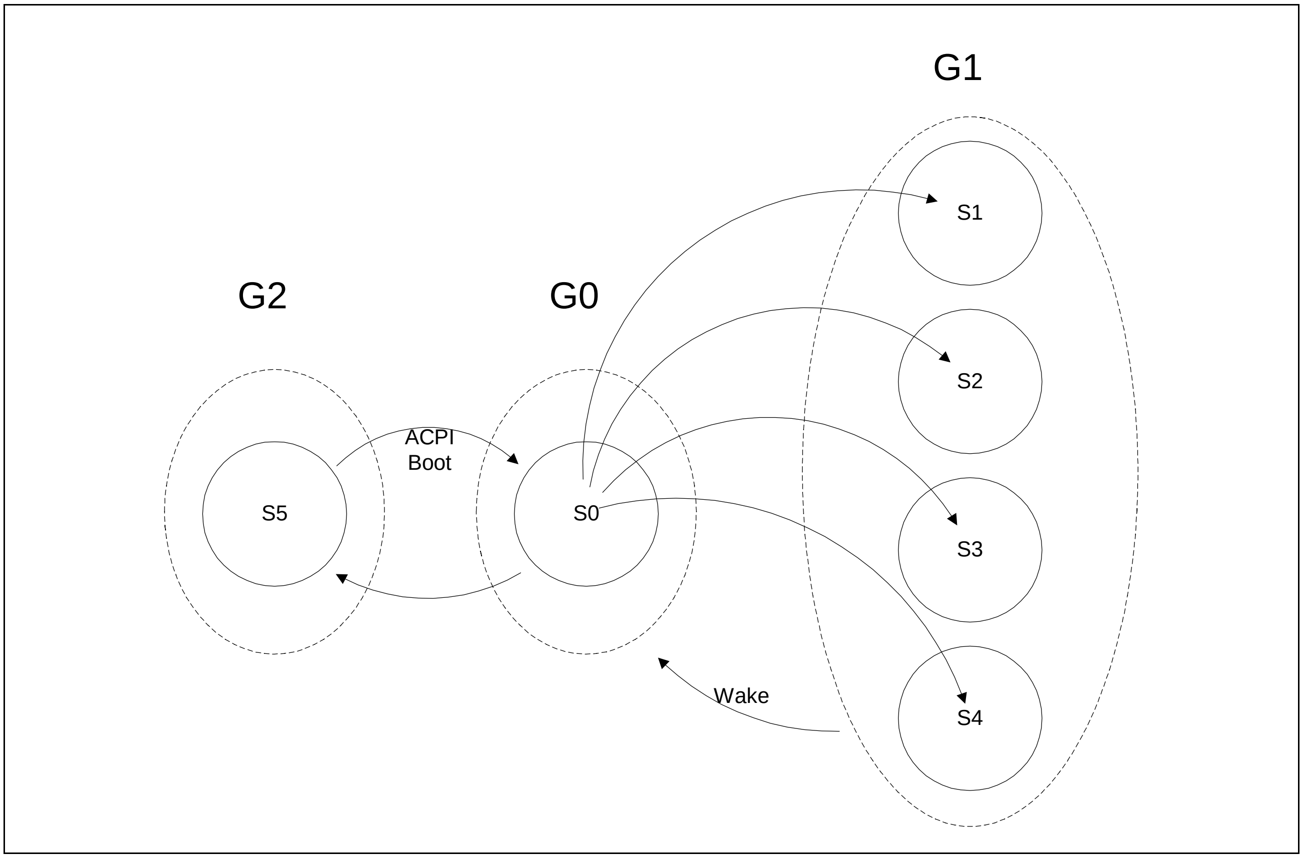


Figure 120: ACPI Sleep States

The ACPI 2.0 specification defines power management behavior for device classes differently. For example, audio device states D0-D3, are not similar to modem device states. Refer to Appendix A of the ACPI 2.0 specification for a detailed discussion of device power management. Table 698 (ACPI 2.0 spec) shows the power management behavior for the network device class.

Table 698: Power Management Behavior for the Network Device Class

State	Implementation Requirement	Definition
D0	Required	Device is powered on and running. Device delivers full functionality.
D1	Optional	No bus transactions allowed. No bus reception allowed. No interrupts can occur. Device context may be lost.
D2	Optional	No bus transactions allowed. No bus reception allowed. No interrupts can occur. Device context may be lost
D3	Required	No bus transactions allowed. No bus reception allowed. No interrupts can occur. Device context is lost.

The power management policy for the network class specifies how a network device behaves at a Dx power state (see Table 699). For example, a device, not the wire-speed of the device, needs to wake up during WOL mode. A network wake up may be initiated by a Magic Packet or similar unique packet format.

Table 699: Power Management Policy for the Network Class

Current State	Next State	Cause/Effect
D0	Dx	• Cause: System enters sleep state S1-S4 • Effect: If wake is enabled, the lowest Dx state is chosen for the network device. There is one limitation, in that the Dx state must support the wakeup.
D0	D3	• Cause: System initiated network shutdown • Effect: System enters sleep state. The device may provide wake capability, but it is not required.
D1/D2/D3	D0	• Cause: System wakeup. • Effect: Device moves to full-power operation.

PCI

PME

The PCI 2.2 specification states:

"The Power Management Event (PME) signal is an optional hardware signal."

Wake capable networking devices will implement PME, so a change in system power management state can be requested. PME must be driven until system software clears the power management event. The PME signal is asynchronous to PCI clocking. Wake events are reported by devices when a Wake_Int bit is set in the PMCSR register. PME is enabled/disabled by setting a Wake_En bit in the PMCSR register. The PMCSR register is not device/application specific.

3.3 VAUX

The PCI 2.2 specification states:

"An optional 3.3 volt auxiliary power source delivers power to PCI add-in cards for generation of power management events when the main power to the card has been turned off by software."

The 3.3 Vaux is pin 14 on Side A of a PCI connector. A device will use Vaux while in a low-power state, like D3.

SLOT POWER

A 3.3V or 5.0V power supply for add-in cards. This is the main power supply for the device while in the full power D0 power management state. Several pins on the PCI connector provide paths to the 5.0V or 3.3V power planes on the host motherboard. Refer to the PCI 2.2 specification for the PCI connector pin-out. The choice and use of slot power is based upon the hardware application.

SMI/SCI

Intel processors support a mode call System Management Mode (SMM). SMM is completely transparent to a native OS and operates in another processor address space. System BIOS may implement event handlers for power management in SMM address space. SMM mode is invoked via a System Management Interrupt (SMI). A generated SMI will cause the following actions:

- The system will enable System Management memory (SMRAM). Normal RAM is disabled.
- Processor registers are saved off to SMRAM.
- Processor registers are initialized with SMM settings.
- Processor jumps to entry point in SMRAM, where the SMI handler is placed.
- Power management registers are queried to determine cause of SMI.
- SMM handler services the appropriate request.
- A Return from System Management Mode (RSM) instruction is issued.
- Processor restores registers saved in step #2.
- Normal RAM is enabled. SMRAM is disabled.

The ACPI specification refers to SMM as a legacy mode and prefers the native O/S (i.e., Microsoft Windows, Linux, etc.) to handle the interrupt. The essential problem with using SMI, resides in moving the processor into SMM mode; the native OS does not have visibility into the event. To solve this design constraint, the ACPI 2.0 specification defines a System Control Interrupt. SCI will notify an OS of a power management event. The SCI is a shared, level triggered interrupt. Figure 121 shows how a system BIOS would configure power management to use either SMM or SCI during the system POST. A SCI_Enable will route power management events to the SCI interrupt arbiter, rather than SMI arbiter.

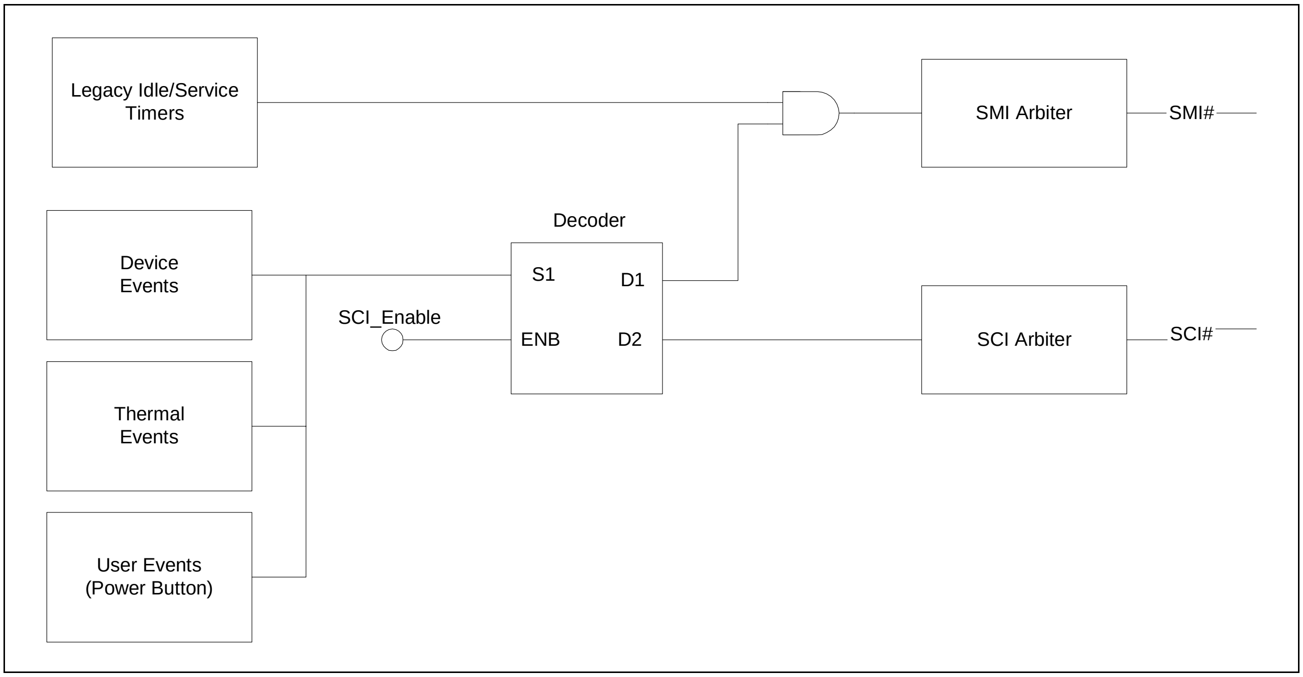


Figure 121: Power Management Configuration During POST

The arbiter is combinational logic residing in the chipset and contains status/control registers for power management. The host chipset provides General-Purpose Event registers (GPE, see Figure 122) to selectively enable/disable PM events. The GPE block also contains status registers to record the PM event's occurrence. All the PM events are mapped into a single-level triggered SCI. The SCI may be routed to either a PIC or APIC. How the SCI is routed is not important for this discussion and will depend upon the host motherboard. The SCI interrupt will be serviced by an ACPI OS, which is the important differentiator. Under SMM mode, the OS would not be notified of the PM event. With an interrupt, the OS will branch to a service routine for the SCI vector. The ACPI driver will have an AML event routine, which will operate upon the PM event accordingly. Refer to the ACPI 2.0 specification for a detailed explanation of how AML and ACPI tables interact. A PME signal can be routed into the GPE block, which ties networking wake-capable devices into the ACPI model.

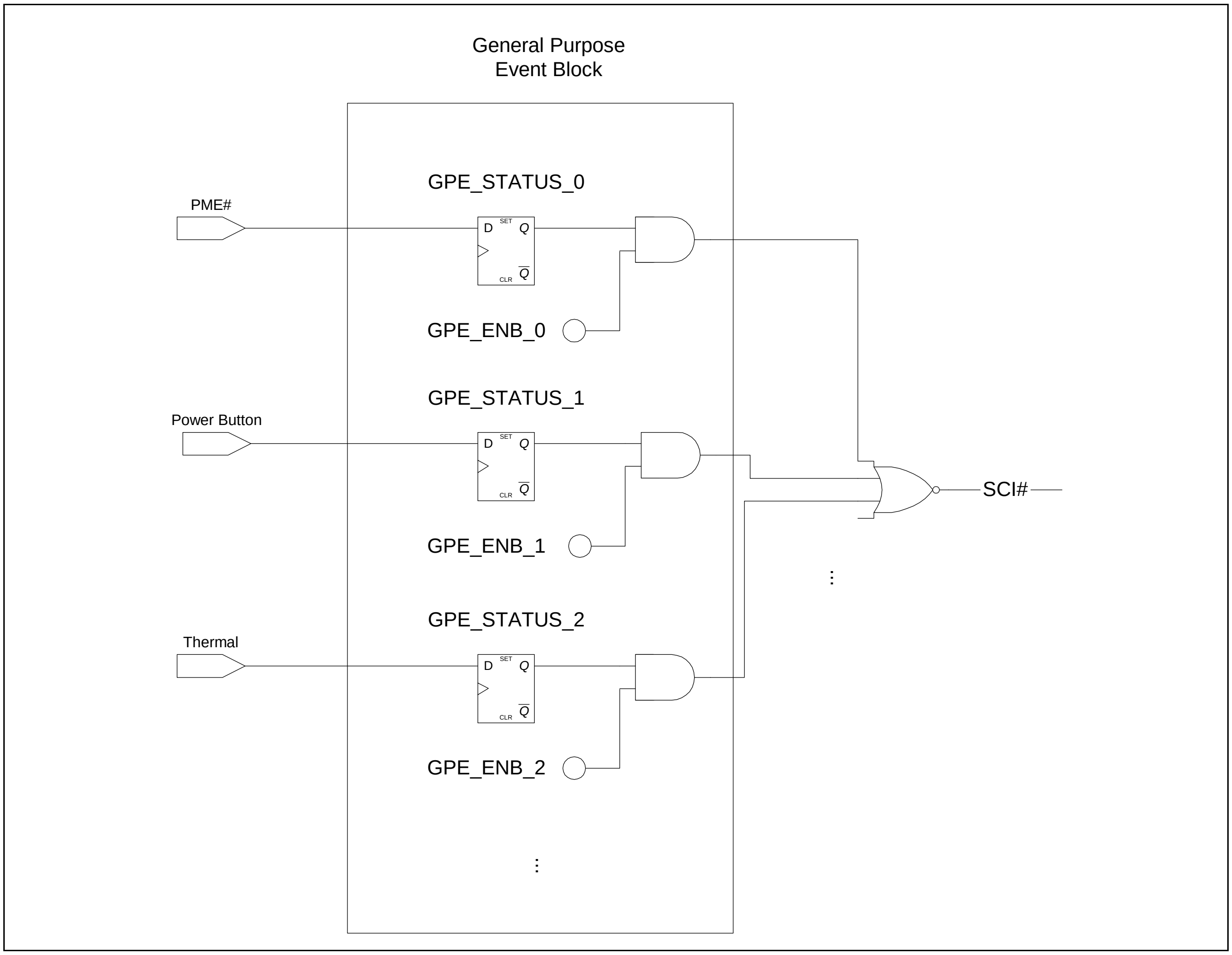


Figure 122: General Purpose Event Block

Appendix C: Initialization and Reset

Signal Sampling to Determine PCI Configuration

Determining Slot Type

The BCM57XX family samples signals $\overline{\text{FRAME}}$, $\overline{\text{IRDY}}$, $\overline{\text{TRDY}}$, $\overline{\text{STOP}}$, and $\overline{\text{DEVSEL}}$ at the rising edge of $\overline{\text{RST}}$ to determine whether the NIC is in a PCI or PCI-X slot (see Figure 123). The chipset drives a PCI-X initialization pattern onto the bus so the NIC can sample the pattern. The combination of $\overline{\text{TRDY}}$, $\overline{\text{STOP}}$, and $\overline{\text{DEVSEL}}$ initializes the PCI-X slot for a specific operating mode. The exact format and permutations of the signals may be found in table 6.2 in the PCI-X v1.0 specification.

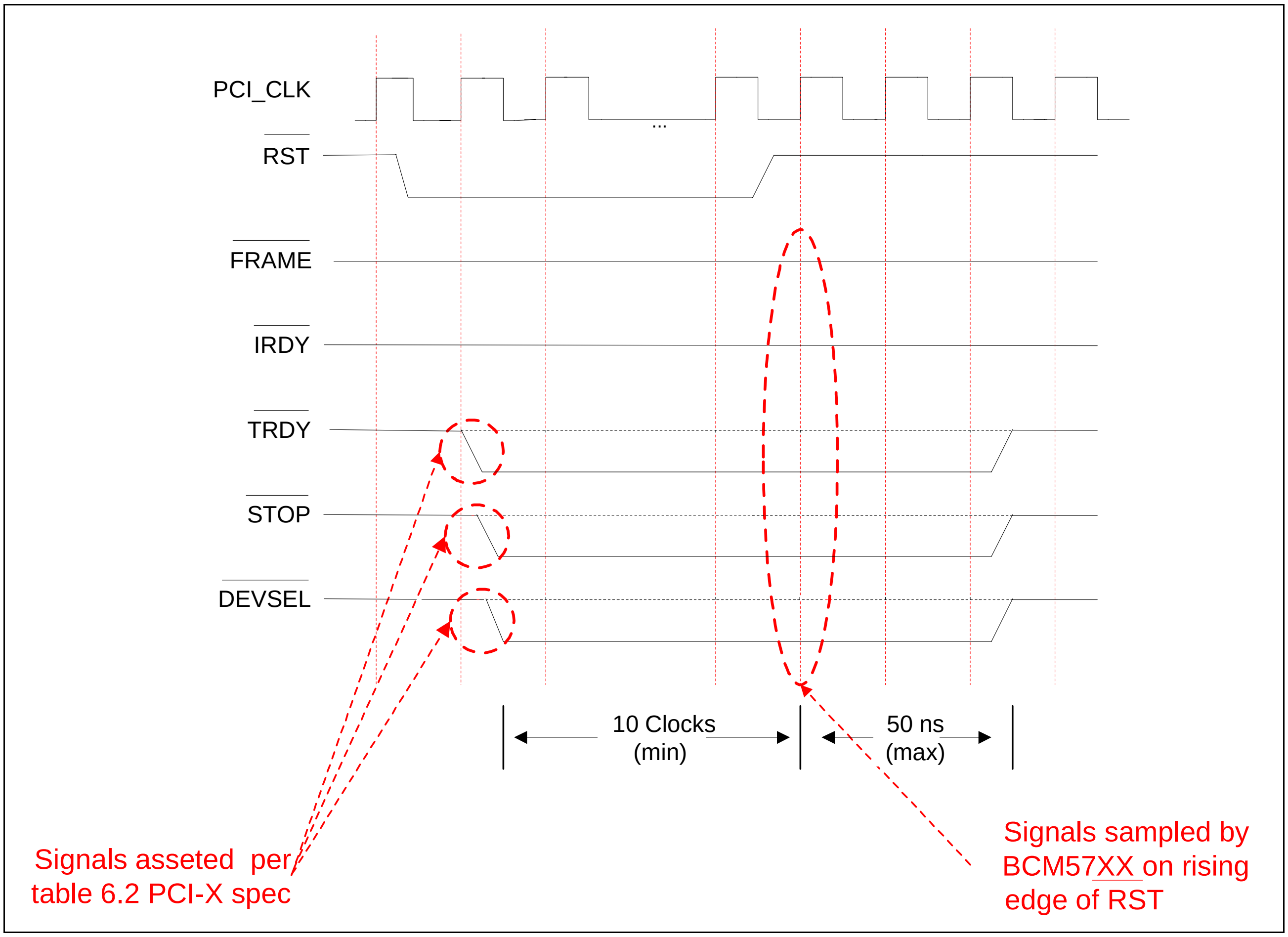

PCI_CLK
$\overline{\text{RST}}$
$\overline{\text{FRAME}}$
$\overline{\text{IRDY}}$
$\overline{\text{TRDY}}$
$\overline{\text{STOP}}$
$\overline{\text{DEVSEL}}$
...
10 Clocks (min)
50 ns (max)
Signals asseted per table 6.2 PCI-X spec
Signals sampled by BCM57XX on rising edge of RST

Figure 123: Sampling Signals to Determine Whether the NIC is in a PCI or PCI-X Slot

Determining PCI Clock

The BCM57XX PCI interface does support 66 MHz conventional PCI operation. The M66EN signal influences the bus frequency for all peripheral devices on the same physical bus. One legacy device may pull the speed of the entire bus down to 33 MHz. The BCM57XX family samples this signal to determine if another peripheral has requested 33-MHz operation (see Figure 124). M66EN is a sense pin on the BCM57XX family. The BCM57XX family will sample this signal after device reset. The PCI chipset will also sample this signal and then scale the PCI clocking accordingly. After the PCI $\overline{\text{RST}}$ signal is deasserted, the BCM57XX family samples the M66EN signal. The M66EN and $\overline{\text{RST}}$ signals are asynchronous to the PCI_CLK. Refer to the PCI v2.2 for exact timing specifications—section 7.5.1. The 33/66 MHz_PCI_66/133 MHz_PCIX bit in the PCI_State register (see "PCI State Register (Offset 0x70)" on page 332) reflects the current bus speed.

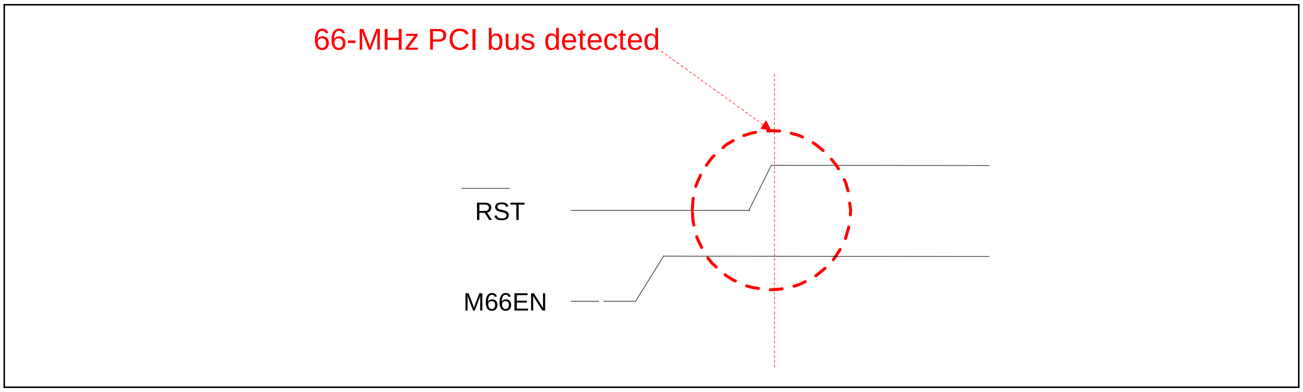


Figure 124: Sampling M66EN to Determine 66 MHz—Start Software PCI Operation

Determining PCI Mode

The BCM57XX family may operate in either 32- or 64-bit PCI mode. When the BCM57XX family comes out of PCI reset, the $\overline{\text{REQ64}}$ signal is sampled (see Figure 125). This signal is asserted to indicate that PCI transactions are 64 bits wide. Refer to section 4.3.2 in the PCI v2.2 specification; the specifics for $\overline{\text{REQ64}}$ setup and hold times are documented. The 32_Bit_PCI_Bus bit in the PCI State register indicates the current sampled status of $\overline{\text{REQ64}}$.

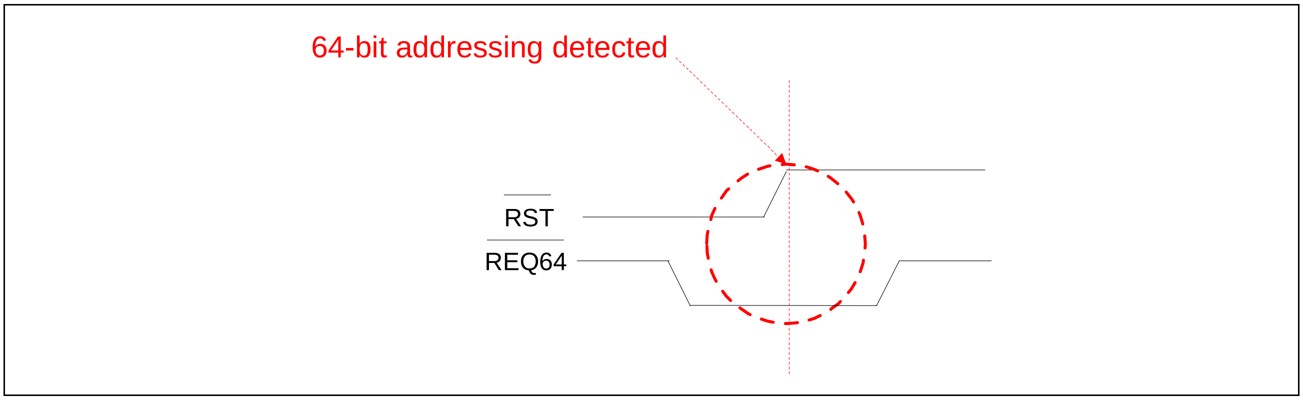


Figure 125: Sampling $\overline{\text{REQ64}}$ to Determine 64-bit PCI Mode

The PCI v2.2 specification states that a device should not be accessed during the 2exp25 clock cycle interval after $\overline{\text{RST}}$ is deasserted (see Figure 126). The BCM57XX family does not target terminate with a retry during this reset interval. PCI transactions should not be initiated since the PCI interface will not assert $\overline{\text{TRDY}}$ and $\overline{\text{FRAME}}$ during the reset interval. However, access to the expansion ROM after the 20 millisecond firmware initialization causes an auto-retry.

Reset Intervals

After the device is reset, ROM code in the device takes 20 ms to 50 ms (depending on the size of the boot firmware) to load the initial firmware code stored in the NVRAM. This initial firmware code then enables the PCI Expansion ROM retry until PXE code is completely loaded from NVRAM to Internal Memory. PXE loading process can take up to 3 or 4 seconds. Expansion ROM auto-retry is removed when PXE code is completely loaded, which only occurs if PXE is administratively enabled. If host software accesses the expansion ROM before the loading process has completed, the device will not target terminate with a retry.

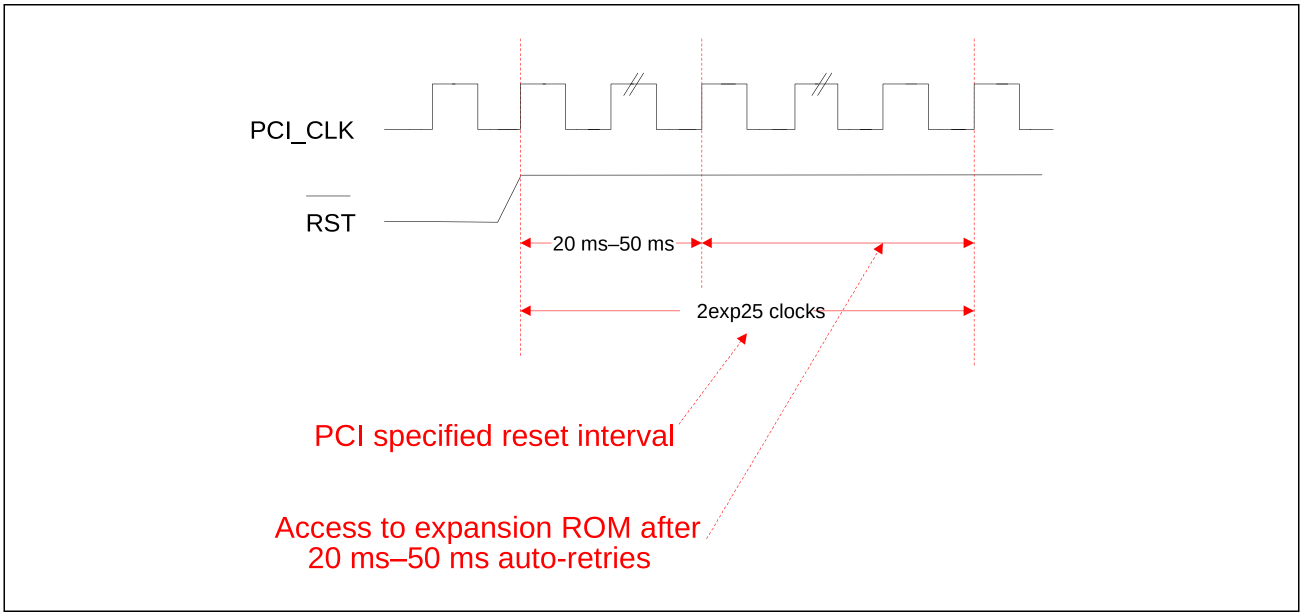


Figure 126: PCI-Specified Reset Interval

GPIO HOLD CONDITION

Figure 127 shows the BCM57XX GPIO hold condition. The GPIO pins hold their current output until the rising edge of reset. If the GPIO pins were tristated before the falling edge of $\overline{\text{RST}}$, the pins will remain tri-stated. After the rising edge of reset, the GPIO pins will tri-state.

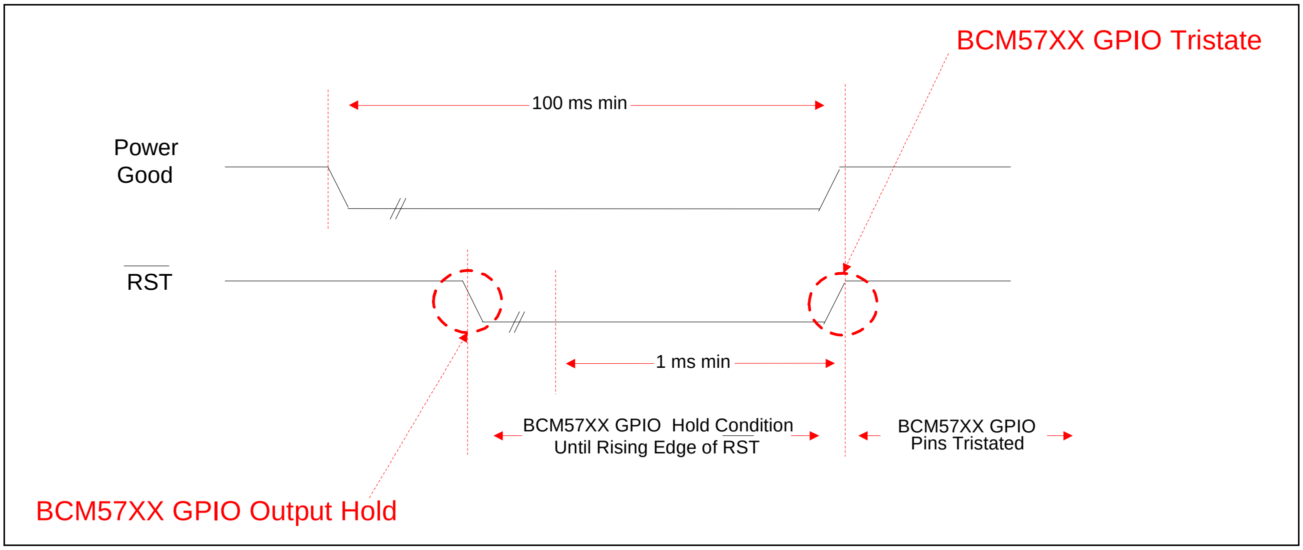

BCM57XX GPIO Tristate
100 ms min
Power Good
RST
1 ms min
BCM57XX GPIO Hold Condition Until Rising Edge of RST
BCM57XX GPIO Pins Tristated
BCM57XX GPIO Output Hold

Figure 127: GPIO Hold Condition

Appendix D: Terminology

Table 700: Terminology

Term	*Definition*
BD	Buffer Descriptor.
Deferred Procedure Call (DPC)	The ISR may schedule a O/S callback to process interrupts at a later time.
DPC	See Deferred Procedure Call (DPC).
Expansion ROM	PCI devices may optionally expose device specific programs to BIOS. For example, network devices may place PXE boot code in their expansion ROM region.
Host Coalescing	A hardware block which the BCM57XX status block. The hardware will drive a line interrupt or MSI.
Interrupt Distribution Queue	The BCM57XX supports four interrupt distribution queues per class of service. The rules engine may place traffic into RX return rings based on rules checking. Within each class of service, the traffic may further be organized in Interrupt Distribution Queues. For example, frames with errors may be given lower data path priority over frames without errors, all within the same class of service (RX Return Ring).
Interrupt Service Routine (ISR)	A procedure where device interrupts are processed.
ISR	See Interrupt Service Routine (ISR).
Pre-boot execution (PXE)	An industry-standard client/server interface that allows networked computers that are not yet loaded with an operating system to be configured and booted remotely.
PXE	See Pre-boot execution (PXE).
Receive BD Initiator	The hardware block that DMA's BDs when receive ring indices are written.
Receive Data and Receive BD Initiator	The hardware block the updates packet buffers, in host memory, after an Ethernet frame is received. The hardware block will also update the BD with information like checksum and VLAN Tags.
Receive Data Completion	The hardware block that updates the host coalescing engine after the packet buffers and BD are DMAed to host memory.
Receive Queue Placement	The hardware block that routes a categorized frame to one of sixteen RX Return rings.
Send BD Initiator	The hardware block that is activated when a Send producer index is updated by host software. The hardware block will DMA a BD from host memory.
Send Data Initiator	The hardware block updates the DMAs in the packet buffers from host memory. The packet buffers are DMAed after the BD has been moved to device local memory.

Broadcom Corporation

5300 California Avenue
P.O. Box 57013
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.