PCI 10/100 Ethernet and V.90/V.92 Controller

440X-PG02-R

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

06/02/09

REVISION HISTORY

Revision	*Date*	*Change Description*
440X-PG02-R	06/02/09	**Updated:** • Table 101: “Transmit Watermark Register Bit Field Descriptions,” on page 92. **Added:** • “LED Control Select Tables” on page 80.
440X-PG01-R	09/12/03	Added changes for BCM4401 B0 revision silicon.
440X-PG00-R	08/23/02	Initial Release.

Broadcom Corporation
5300 California Avenue
Irvine, CA 92617

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Section 1: Overview

BCM440X Description

The BCM440X is a fully integrated 10/100 Ethernet MAC/PHY and V.90/V.92 modem (BCM4402 only) device targeted for use in PCI/Mini PCI adapters, and LAN on Motherboard (LOM) implementations. The BCM440X controller provides a PCI 2.2 and Mini PCI 1.0 host interface. Complete WHQL certified drivers for Windows® 98, Windows Millennium Edition (ME), Windows 2000, and Windows XP are available for 10/100 Ethernet and V.90/V.92 data/fax/voice (BCM4402 only) functions. Each interface operates concurrently as a separate function.

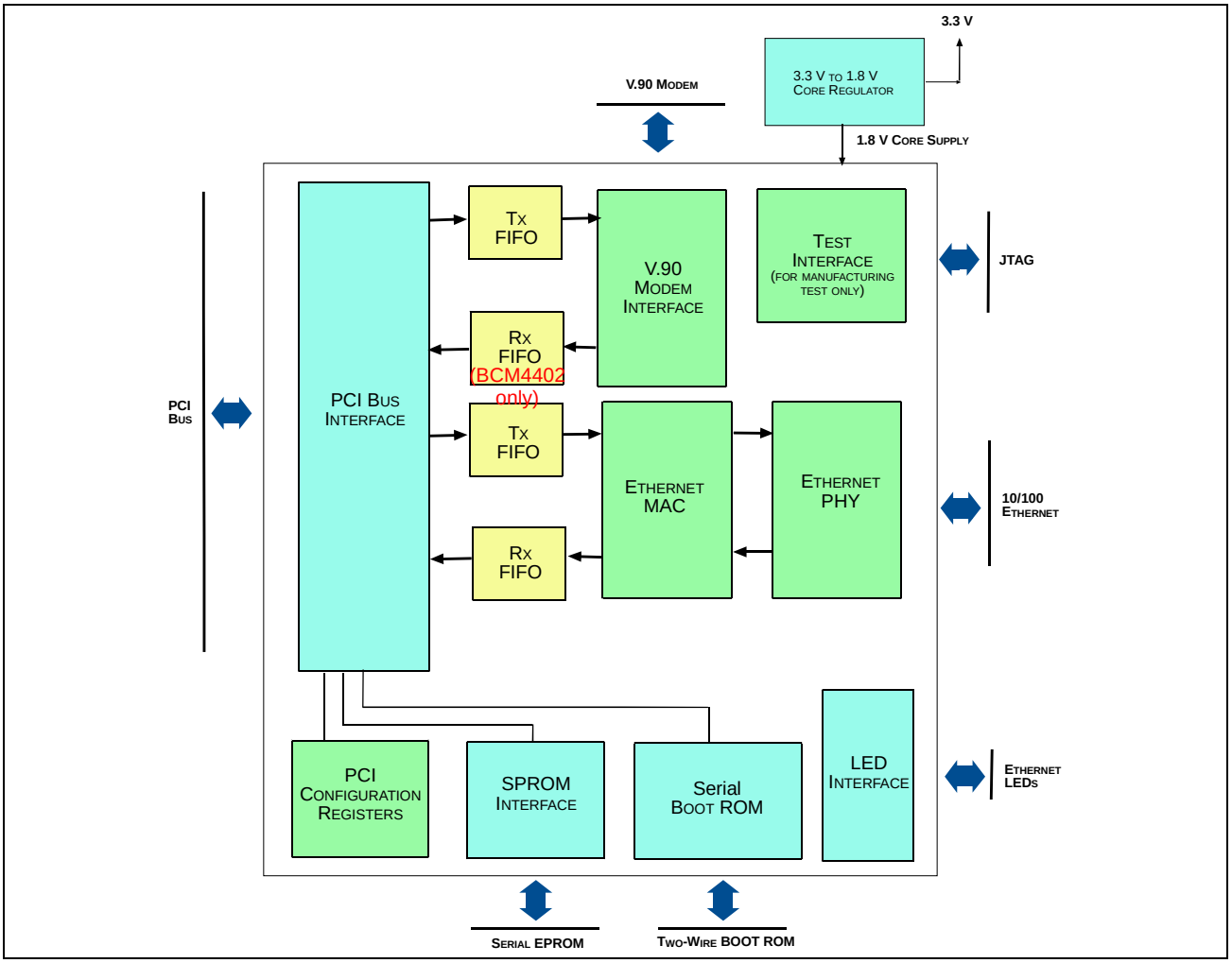

Figure 1: BCM440X Block Diagram

BCM440X FEATURES

- Combination 10/100 Ethernet MAC/PHY and PCI Soft Modem with concurrent operation
- PCI 2.3, Mini PCI 1.0, and LOM
- Meets PCI Power Management Interface Specification v1.1 for Advanced Power Management
- Fully PC2001 WHQL certified drivers for Windows 98, Windows Millennium Edition, Windows 2000, and Windows XP

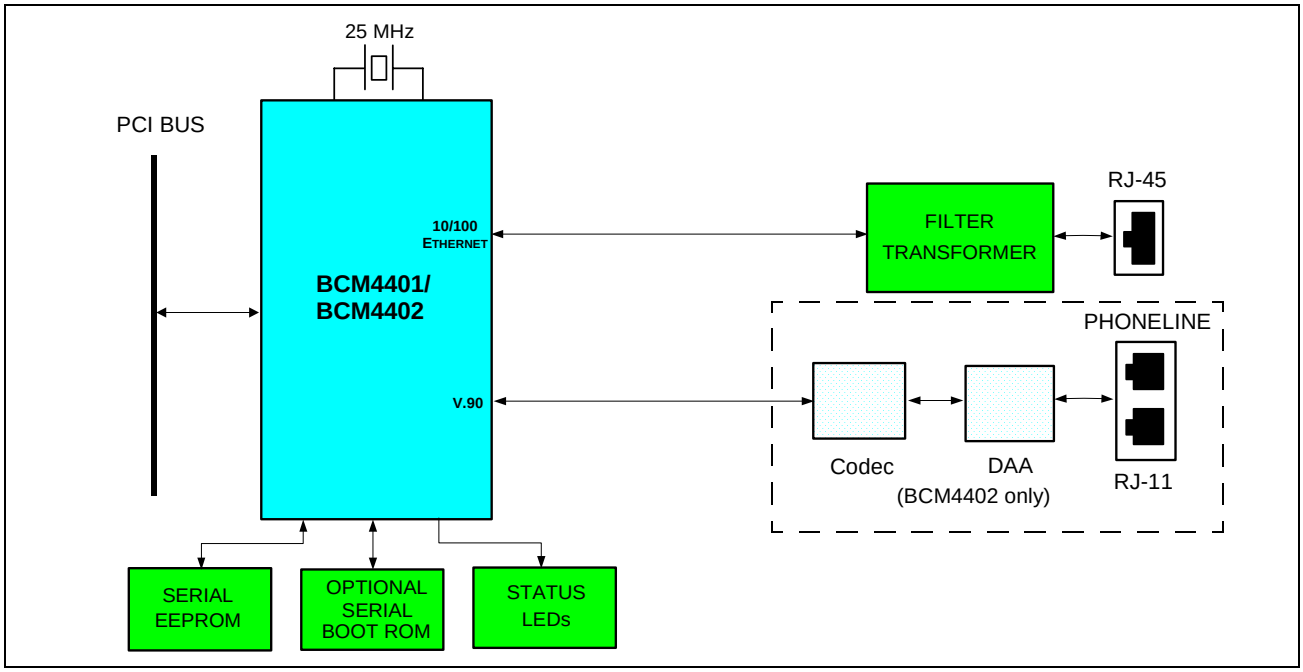

Figure 2: BCM440X System Diagram

- 1 KB, 4 KB, or 16 KB serial EEPROM support for MAC address and configuration parameters
- Optional serial boot ROM support for PXE support
- JTAG
- 0.18µ CMOS, 3.3V supply, 3/5V PCI I/O
- Integrated 3.3V to 1.8V regulator
- 33 MHz PCI 2.2 Local Bus
- 128-pin LQFP

10/100 ETHERNET MAC/PHY

The BCM440X 10/100 Ethernet function incorporates a complete IEEE 802.3u compliant MAC and PHY. The MAC also supports management functions including RMON, Ethernet-like MIB, and IEEE 802.3u Ethernet MIB. The 10/100 PHY uses Broadcom's advanced Digi-Φ™ architecture for low power and includes advanced power management. The 10/100 PHY also supports Auto-Negotiation and HP Auto-MDIX functions to eliminate installation errors.

- IEEE 802.3u compliant Transmit and Receive engines
- Transmit and Receive FIFOs (2KB transmit and 4KB receive)
- Full- and Half-Duplex operation
- Full-duplex frame-based Flow Control compliant with IEEE 802.3x
- Internal 64-entry Content Addressable Memory (CAM) for perfect address filtering
- MII Management interface allows control of internal transceiver functions
- Internal MAC and PHY loopbacks
- Integrated MIB counters supporting RMON, Ethernet-like MIB, and IEEE 802.3™ Ethernet MIB
- Low power 10BASE-T/100BASE-TX IEEE 802.3u Fast Ethernet Transceiver
- Auto-Negotiation support with Next Page
- Baseline Wander correction
- Link LEDs (Link, Activity, Speed, Collision)
- Automatic Power-Down mode
- Unique Energy Detection Circuit enabling Intelligent Power Management
- HP Auto-MDIX
- Cable Length Indication
- Cable lengths greater than 120 meters supported

CODEC MODEM FUNCTION (BCM4402 ONLY)

The BCM440X controller includes a data/fax modem interface to an external serial Modem Codec/DAA, allowing implementation of a complete, low-cost data/fax/voice soft modem. The data/fax Software Modem driver is available for Windows 98, Windows Millennium Edition, Windows 2000, and Windows XP.

DEVICE INITIALIZATION AND RESET

Refer to the following sections for device initialization and reset:

- Section 2: "Backplane":
- Section 3: "DMA": see "Descriptor Processor Channel Initialization" on page 45 and "Descriptor Processor Channel Reset" on page 45.
- Section 4: "Codec Core (BCM4402 only)":
- Section 5: "Ethernet 10/100 Core": see "Ethernet Core Initialization" on page 162.
- Section 6: "PCI Core":
- Section 7: "Programming Hints": see "Reset" on page 203

RELATED DOCUMENTS

- BCM4401 or BCM4402 data sheet
- Codec data sheet (from codec vendor)
- DAA data sheet (from DAA vendor)

Section 2: Backplane

BACKPLANE, CORES, AND AGENTS

The BCM440X uses a system backplane to connect all of its functional blocks. These functional blocks, known as cores, use the Open Core Protocol (OCP) interface to communicate with agents attached to the system backplane.

The BCM440Xis a member of a family of chips built using the same system backplane architecture. Each member of the family contains a different set of cores, but shares basic architectural features such as address space definition, interrupt and error architecture, and backplane register definitions. The information contained in this chapter is applicable to the entire family of chips and can contain information that is not relevant to the BCM440X.

In this chapter, the term *host interface core* refers to cores that implement slave devices that are accessible to outside hosts. The PCI and PCMCIA cores are host interface cores. *Communications cores* are those that implement networking protocols such as InsideLine or Ethernet.

The BCM440X contains three cores:

- Ethernet MAC
- Codec/V.90 (not applicable to the BCM4401)
- PCI

The BCM440X also contains an Ethernet PHY that is accessed using the Ethernet MAC core.

Each core can have an initiator agent that passes read and write requests onto the system backplane and a target agent that returns responses to those requests. Not all cores contain both an initiator and a target agent. Initiator agents are present in cores that contain host interfaces (PCI, PCMCIA), embedded processors (MIPS), or DMA processors associated with communications cores. All cores other than PCMCIA have a target agent.

SUPPORTED OPERATIONS

The system backplane defines four types of bus transactions: reads, writes, exclusive reads, and broadcast writes. The BCM440X does not implement exclusive reads. Broadcast writes are used only to update system backplane configuration registers and are performed by accessing the PCI core's BroadcastAddr register and then writing the desired value to the PCI core's BroadcastData register.

ARBITRATION

The system backplane arbitration mechanism consists of a high-priority Time-Division Multiplexing (TDM) allocation and a lower-priority token passing ring. Allocated TDM slots give cores a right of first refusal for system backplane accesses during that slot. By default, all cores compete for the round robin token. All of the other cores compete for the round robin token.

If necessary, the SBBWA0 configuration register (see "System Backplane Bandwidth Allocation Table 0 Register (SBBWA0, offset 0xFA0)" on page 16) and the SBPOLICY field of the SBIMSTATE configuration register (see "System Backplane Initiator State Register (SBIMSTATE, offset 0xF90)" on page 12) in the cores can be used to adjust system backplane arbitration priority.

In the BCM440X, the arbitration scheme cannot be changed by register accesses. TDM slots are not used, and all cores compete for the round robin token.

INTERRUPT ARCHITECTURE

Each interrupt-generating core collects interrupt conditions internally and outputs a single interrupt signal over one of the sbFlags system backplane signals. sbFlags[2:0] are allocated to cores that raise interrupts as shown in Table 1.

Table 1: Interrupt Cores

Bit	*Core*
0	PCI
1	Ethernet MAC
2	Codec

The core target agents do not use the Interrupt signal to report interrupts. As a result, the SBINTMODE and SBINTFLG fields of the SBTMCONFIGLOW registers (see "System Backplane Target Configuration Low Register (SBTMCONFIGLOW, offset 0xFB8)" on page 20 and "System Backplane Target Configuration High Register (SBTMCONFIGHIGH, offset 0xFBC)" on page 21) are not used. Instead, cores use sFlag[0] to report interrupts and use the SBTTPSFLAGNUM0 field in the SBTPSFLAG register (see "System Backplane Target OCP Slave Flag Control Register (SBTPSFLAG, offset 0xF18)" on page 10) to route the interrupt to the correct sbFlag in the system backplane.

ERROR ARCHITECTURE

There are three types of errors that occur on the system backplane: in-band error responses to read requests, out-of-band error assertions in response to write requests, and time-outs.

The system backplane supports two error reporting mechanisms: in-band error responses and an out-of-band shared error signal, SBERROR. Configuration register settings control which errors are reported on SBERROR and which cores receive SBERROR indications. In addition, each initiator and target agent contains error configuration and logging registers.

A read request causes an error response when it times out, when the target address is unclaimed by any core, and when the address is an undefined address within a core's address space. Time-outs are logged in the SBTOERR field of the SBIMSTATE register (see "System Backplane Initiator State Register (SBIMSTATE, offset 0xF90)" on page 12) and the other errors are logged in the SBIBERR field. In all cases, an error response is returned to the initiating core.

Because all writes on the system backplane are posted, there is no mechanism for in-band error indications to the initiating core. An attempt to write to an unclaimed address is logged in the SBIBERR field of the initiating agent and reported to the initiating core. Request time-outs are logged in the SBTOERR field. In response to these errors, the MIPS core generates a non-maskable interrupt and the PCI core and DMA engines log the error and assert an error interrupt.

Note: The setting of the InBandErrMode field also affects the reporting of read errors.

For write errors detected in the target, the initiating core does not receive any direct error indication. In the default configuration, out-of-band errors reported by any core assert the SBERROR signal. The SBERROR signal is sent to the MIPS core where it causes a non-maskable interrupt. Driver software must traverse the target cores examining error log bits to identify and clear the source of the out-of-band errors. Cores can not respond to system backplane accesses when the Reset or Reject fields of the SBTMSTATELOW register (see "System Backplane Target State Low Register (SBTMSTATELOW, offset 0xF98)" on page 14) are set or when the ClockEnable field of the SBTMSTATELOW register is clear.

ADDRESS SPACE MAP

The BCM440X address space is described in Table 2. The addresses of all on-chip resources are restricted to the first 512 MB so that they can be accessed from each of the MIPS *X* segments, giving software maximum flexibility to determine the mapability and cacheability of each reference.

Table 2: BCM440X Address Space

Address Range		Contents	Size
0x0000_0000	0x07FF_FFFF	PCI	128 MB
0x0000_0000[a]	0x07FF_FFFF	SDRAM	128M
0x0800_0000	0x0FFF_FFFF	PCI	128M
0x1000_0000[b]	0x17FF_FFFF	SDRAM byte-swap	128 MB
0x1800_0000[a]	0x1800_0FFF	Ethernet MAC registers	4k
0x1800_1000[b]	0x1800_1FFF	Codec registers (BCM4402 only)	4k
0x1800_2000[b]	0x1800_2FFF	PCI registers	4k
0x1800_3000[b]	0x1800_FFFF	Reserved	–
0x1900_0000	0x1EFF_FFFF	Reserved	–
0x1F00_0000[a]	0x1F0F_FFFF	PCMCIA memory accesses	1 MB
0x1F10_0000[a]	0x1F1F_FFFF	PCMCIA I/O accesses	1 MB
0x1F20_0000[a]	0x1F2F_FFFF	PCMCIA configuration accesses	1 MB
0x1F30_0000[a]	0x1F7F_FFFF	Reserved	–
0x1F80_0000[a]	0x1FBF_FFFF	Configurable External Interface	4 MB
0x1Fc0_0000[a]	0x1FFF_FFFF	Flash	4 MB
0x4000_0000	0x7FFF_FFFF	PCI	1 GB

a. Not supported in BCM440X.
b. Enumeration Space.

All resources are mapped in one of these six address regions: SDRAM space, PCI space, big-endian SDRAM space enumeration space, core-dependent space, and external interface space.

The lower part of the address space is used to access up to 128MB of external SDRAM. The next 128MB region supplies a 128MB window onto an external PCI space. See Section 6: "PCI Core" for information on mapping internal addresses to PCI addresses. The PCI address space can be remapped dynamically to another otherwise unused address range. In particular, it can be remapped to the range starting at 0 to support systems that lack SDRAM. In this case, the SDRAM address region must be disabled.

The third 128MB region provides a big-endian window on to SDRAM space. Write accesses to this region are byte-swapped before being written to SDRAM and read accesses are byte-swapped before being returned to the system backplane.

Each core supports a 4KB internal register interface space that contains configuration control and status registers and, in most cases, the complete programming interface for the core. The configuration block occupies the last 256 bytes of these spaces.

These register interface spaces are allocated contiguously in the region designated as enumeration space in Table 2 on page 7. Because the register interface spaces are contiguous and contain configuration information at known offsets, software can determine which cores a present by reading the SBIDHIGH register (see "System Backplane Identification High Register (SBIDHIGH, offset 0xFFC)" on page 26) in each register space starting at 0x18000FFC and incrementing by 4KB until an invalid address is encountered. The SBIDHIGH register contains identifying information for its core. The address space after the end of the external device region is reserved for cores that require more than 4KB for their programming models.

CORE CONFIGURATION SPACE REGISTERS

Each core contains a configuration block at offset 0xF00 in the host interface space. Registers in this block are used to identify cores, define address spaces, control clocks and reset, and setup arbitration, interrupt, and error routing information. The register definitions and many characteristics and reset values are common across all cores, but there are differences between core configurations that are described below.

These registers are part of the agent rather than the core. In particular, they are not affected by the core reset signals.

Table 3 shows the configuration registers that are present in at least one core.

Table 3: Core Configuration Space Registers

Address Offset	*Name*	*Abbreviation*	*Type*	*Buffered*	*See...*
0xF08	Initiator OCP slave flag control[a]	SBIPSFLAG	R/W	No	9
0xF18	Target OCP slave flag control[a]	SBTPSFLAG	R/W	No	10
0xF60	Address match 3[a]	SBADMATCH3	R/W	Yes	10
0xF68	Address match 2[a]	SBADMATCH2	R/W	Yes	10
0xF70	Address match 1[a]	SBADMATCH1	R/W	Yes	11
0xF90	Initiator state	SBIMSTATE	R/W	No	12
0xF94	Interrupt vector	SBINTVEC	R/W	No	13
0xF98	Target state low	SBTMSTATELOW	R/W	No	14
0xF9C	Target state high	SBTMSTATEHIGH	R/W	No	15
0xFA0	Bandwidth allocation table 0	SBBWA0	R/W	Yes	16
0xFA8	Initiator configuration low	SBIMCONFIGLOW	R/W	Yes	17
0xFAC	Initiator configuration high	SBIMCONFIGHIGH	R/W	Yes	18
0xFB0	Address match 0	SBADMATCH0	R/W	Yes	19
0xFB8	Target configuration low	SBTMCONFIGLOW	R/W	Yes	20
0xFBC	Target configuration high	SBTMCONFIGHIGH	R/W	Yes	21
0xFC0	Broadcast configuration register	SBBCONFIG	R/W	Yes	22
0xFC8	Broadcast state register	SBBSTATE	R/W	No	23
0xFD8	Activate configuration	SBACTCNFG	pseudo	No	24
0xFE8	Flag status[a]	SBFLAGST	R	No	24
0xFF8	Identification low	SBIDLOW	R	No	25
0xFFC	Identification high	SBIDHIGH	R	No	26

a. Not included in all cores.

UPDATING BUFFERED REGISTERS

Registers designated as buffered are designed to be updated simultaneously in all agents to guarantee proper SB operation. In order to update a buffered register, software must perform writes to all cores where the register's value changes and then must perform a broadcast write to the SBACTCNFG pseudo-register (see "System Backplane Activate Configuration Register (SBACTCNFG, offset 0xFD8)" on page 24) to commit the previous writes. This write is executed by writing the value 0xFD8 to the PCI core's BroadcastAddress register (see "Broadcast Address Register (BroadcastAddress, offset 0x50)" on page 173) and then writing a 0 to the PCI Core's BroadcastData register (see "Broadcast Data Register (BroadcastData, offset 0x54)" on page 173).

If subsequent operations must wait for the buffered register update to complete, software should perform another access to a register in the PCI core. That access does not complete until after the broadcast write completes. None of the buffered registers on the BCM440X contain writable fields, so these procedures are not necessary.

SYSTEM BACKPLANE INITIATOR OCP SLAVE FLAG CONTROL REGISTER (SBIPSFLAG, OFFSET 0XF08)

The four-byte SBIPSFLAG register is used to route backplane flags to cores that handle interrupts. Each core that is capable of signaling an interrupt has an associated backplane flag. For cores that do not handle interrupts, this register is read-only and contains the value 0x0. For the MIPS core, this register is writable and maps interrupts from selected cores onto MIPS interrupts 1 through 4.

(For BCM4401 B0 only) For the PCI core, this register is read-only and is initialized so that input flag N is the interrupt used by the Nth core in enumeration space. See "PCI Interrupt Status Register (PCIIntStatus, offset 0x90)" on page 196 for more information on routing interrupts to the PCI bus.

Table 4: System Backplane Initiator OCP Slave Flag Control Register Bit Field Descriptions

Field	Name	Description	Access	Default
31:30	Reserved	–	–	–
29:24	IntFlag4 (F4)	Interrupt Flag 4 This field selects which core interrupt is routed to interrupt 4.	–	–
23:22	Reserved	–	–	–
21:16	IntFlag3 (F3)	Interrupt Flag 3 This field selects which core interrupt is routed to interrupt 3.	–	–
15:14	Reserved	–	–	–
13:8	IntFlag2 (F2)	Interrupt Flag 2 This field selects which core interrupt is routed to interrupt 2.	–	–
7:6	Reserved	–	–	–
5:0	IntFlag1 (F1)	Interrupt Flag 1 This field selects which core interrupt is routed to interrupt 1.	–	–

See "System Backplane Interrupt Vector Register (SBINTVEC, offset 0xF94)" on page 13 for information on routing multiple core interrupts to MIPS interrupt 0. See "Interrupt Architecture" on page 6 for more information on interrupts.

System Backplane Target OCP Slave Flag Control Register (SBTPSFLAG, offset 0xF18)

The four-byte SPTPSFLAG register is read-only and is used for interrupt routing. The entire register contains 0 for cores that do not generate interrupts.

Table 5: System Backplane Target OCP Slave Flag Control Register Bit Field Descriptions

Field	Name	Description	Access	Default
31:7	Reserved	–	RO	0
6	SBTPSFLAG0EN0 (FE)	Target OCP Slave Flag 0 Enable 0 This field is set to 1, indicating that the interrupt is always sent on the system backplane.	RO	
5:0	SBTPSFLAGNUM0 (FN)	Target OCP Slave Flag Number 0 This field specifies the number of the sbFlag system backplane flag used for the interrupt generated by this core. It is set to the values shown in Table 6.	RO	See Table 6

Table 6: sbFlag Values

Flag Number	Core
0	PCI
1	Ethernet MAC
2	Codec
3-63	Reserved

The remaining fields, SBTPSFLAG0ENx and SBTPSFLAGNUMx, where x is non-zero, always contain zero.

See "System Backplane Interrupt Vector Register (SBINTVEC, offset 0xF94)" on page 13 and "System Backplane Initiator OCP Slave Flag Control Register (SBIPSFLAG, offset 0xF08)" on page 9 for information on how interrupts are routed from the system backplane flags to cores that handle interrupts.

System Backplane Address Match 3 Register (SBADMATCH3, offset 0xF60)

See "System Backplane Address Match 1 Register (SBADMATCH1, offset 0xF70)".

System Backplane Address Match 2 Register (SBADMATCH2, offset 0xF68)

See "System Backplane Address Match 1 Register (SBADMATCH1, offset 0xF70)".

SYSTEM BACKPLANE ADDRESS MATCH 1 REGISTER (SBADMATCH1, OFFSET 0XF70)

These four-byte SBADMATCHx registers are used by some cores to define additional address spaces beyond the standard 4KB host interface space defined by the SBADMATCH0 register. Use of these registers is described in the OCP Interface section of the documentation for each core that requires an additional address space.

Table 7: System Backplane Address Match Registers Bit Field Descriptions

Field	*Name*	*Description*	*Access*	*Default*
31:8 (Type 0) 31:12 (Type 1) 31:16 (Type 2)	AddressBase (AB)	Address Base This field contains the base address for this range.	–	–
11 (Types 1 and 2 only)	NegativeMatch (NM)	Negative Match When this field is set to 1, this address match register performs subtractive decoding, accepting all addresses that do not match the range specified by the AddressBase and AddressSize registers. When cleared to 0, positive address matching is used.	–	–
10 (Types 1 and 2 only)	AddressEnable (AE)	Address Enable When this field is set to 1, the address matching performed by this register is enabled. When cleared to 0, the address range specified by this register is disabled. Type 0 address ranges are always enabled.	–	–
7:3 (Type 0) 8:3 (Types 1 and 2)	AddressSize (AS)	Address Size This field specifies the size of the address region. It contains the index of the most-significant address bit that does not participate in the match. System backplane address bits [31:AddressSize+1] are compared to AddressBase[31:AddressSize+1] to determine whether the address matches the range specified by this register. This field is read only for address types 0 and 1 and has a minimum value of 15 for type 2.	–	–
1:0	AddressType (AT)	Address Type This read-only field specifies the kind of address matching performed by this register. If this field contains 0, the address register performs a positive decode on a fixed range. Address types 1 and 2 can perform either positive or subtractive decodes, and address type 2 supports a programmable region size.	–	–

SYSTEM BACKPLANE INITIATOR STATE REGISTER (SBIMSTATE, OFFSET 0XF90)

The four-byte SBIMSTATE register is writable and contains the value 0x0 for the MIPS core and 0x20 for all other cores on device reset. This register contains information about the core's use of the system backplane bus.

Table 8: System Backplane Initiator State Register Bit Field Descriptions

Field	Name	Description	Access	Default
31:19	Reserved		R/W	–
18	TimeOut (TO)	Timeout This field is set to 1 when a bus transaction initiated by this core does not receive a response before the timeout period elapses. The BCM440X never clears this field; it can only be cleared by writing it to 0. Writing a 1 to this field has no effect.	R/W	–
17	InBandError (IE)	InBand Error This field is set to 1 when a bus transaction initiated by this core receives an error response. The BCM440X never clears this field; it can only be cleared by writing it to 0. Writing a 1 to this field has no effect.	R/W	–
16:6	Reserved	–	R/W	–
5:4	Policy (PL)	Policy This field controls the arbitration policy for the core as shown in Table 9.	R/W	–
3:0	PipeCount (PC)	Pipe Count This read-only field contains the value 0, disabling system backplane request throttling.	R/W	–

Table 9: Arbitration Policy

Value	Arbitration Policy
0	Use both time slices and token.
1	Use time slices only.
2	Use token only.
3	Reserved.

SYSTEM BACKPLANE INTERRUPT VECTOR REGISTER (SBINTVEC, OFFSET 0XF94)

The four-byte SBINTVEC register contains the value 0x0 on device reset. This register is not used, and is not writable, for cores that do not handle interrupts.

The register is writable and bits of this register enable interrupts from a specified core to assert a core-dependent interrupt signal. Table 10 shows which core interrupt is enabled by each bit in this register.

Table 10: System Backplane Interrupt Vector Register Core Interrupts

Bit	*Interrupt Enabled For*	*Access*	*Default*
31:3	Reserved		0
2	Codec		0
1	Ethernet MAC		0
0	PCI		0
	Reserved (BCM440 B0 only)	–	–

Note: By default, the USB, Ethernet MAC, InsideLine, and PCI interrupts are routed to MIPS interrupts 1 through 4. See "System Backplane Initiator OCP Slave Flag Control Register (SBIPSFLAG, offset 0xF08)" on page 9 and "Interrupt Architecture" on page 6 for more information on interrupt handling.

SYSTEM BACKPLANE TARGET STATE LOW REGISTER (SBTMSTATELOW, OFFSET 0XF98)

The four-byte SBTMSTATELOW register is writable and contains a core-dependent value on device reset. This register controls several test and reset functions for the core. The reset value is 0x10000 for the processor, the host interface cores, and the external interface core. For all other cores, the reset value is 0x1.

Table 11: System Backplane Target State Low Register Bit Field Descriptions

Field	Name	Description	Access	Default
31	BISTEnable (BE)	BIST Enable When set to 1, this field initiates the BIST (Built-In Self Test) cycle for all RAMs in this core. This field should be cleared to 0 after the BISTDone and BISTFail status fields have been read from the SBTMSTATEHIGH register. The ClockEnable and ForceGatedClocksOn fields must be set when running BIST, and reset must be cleared.	R/W	–
30	PMEEnable (PE)	PME Enable When set to 1, this field enables the generation of power management events by this core. When cleared to 0, power management event generation is disabled.	R/W	–
29:18	Flags (FL)	Flags This field is available to individual cores for core-specific purposes. Documentation for these flags can be found in the chapters describing cores that use them.	R/W	–
17	ForceGatedClocksOn (FC)	Force Gated Clocks On When this field is set to 1, all core clocks run regardless of current activity. When this field is cleared to 0, clocks in inactive portions of the core are turned off to conserve power. Writing this field has no effect in cores that do not implement gated clocks.	R/W	–
16	ClockEnable (CE)	Clock Enable When this field is cleared to 0, the core clock is disabled for maximum power savings. This field must be set to 1 for normal operation.	R/W	–
15:2	Reserved	–	R/W	–
1	Reject (RJ)	Reject When set to 1, this field causes the core to reject all transactions to non-configuration registers. This field should be set to 1 when the core's clocks are disabled and set to 0 at all other times.	R/W	–
0	Reset (RS)	Reset When set to 1, this field resets all state in the core and clears state associated with pending transactions from this core. When changing the value of this field, software should ensure that the ClockEnable and ForceGatedClocksOn fields are set to allow reset to propagate to the entire core. See "Reset" on page 203 for more information on resetting cores. Software should also set the ForceGatedClocksOn and ClockEnable fields when setting this field.	R/W	–

SYSTEM BACKPLANE TARGET STATE HIGH REGISTER (SBTMSTATEHIGH, OFFSET 0XF9C)

The four-byte SBTMSTATEHIGH register is writable and contains the value 0x0 on device reset. This register contains core status information.

Table 12: System Backplane Target State High Register Bit Field Descriptions

Field	*Name*	*Description*	*Access*	*Default*
31	BISTDone (BD)	BIST Done This field is set to 1 when the core's Built-In Self Test completes, and is cleared to 0 when software clears the BISTEnable field of the SBTMSTATELOW register.	–	0
30	BISTFail (BF)	BIST Fail This field is set to 1 when the core's Built-In Self Test detects an error. This field is valid only when the BISTDone field is set to 1. The BISTStatus register in the core contains more information on the failure.	–	0
29	GatedClkRequest (GC)	Gated Clock Request This field is set to 1 when the core is requesting that its gated clocks be turned on. This field is clear when there is no activity that requires the gated clocks to be running. This field is not present in cores that do not have gated clocks.	–	0
28:3	Reserved	–	–	0
2	Busy (BY)	Busy This field is set to 1 when the core is processing a bus transaction and is set to 0 when it is not.	–	0
1	Interrupt (IN)	Interrupt This field is set to 1 when the core is asserting an interrupt and is set to 0 when it is not.	–	0
0	SError (SE)	Slave Error This field is set to 1 when a bus transaction received by this core causes an SError response. The device never clears this field; it can only be cleared by writing it to 0. Writing a 1 to this field has no effect.	–	0

SYSTEM BACKPLANE BANDWIDTH ALLOCATION TABLE 0 REGISTER (SBBWA0, OFFSET 0XFA0)

The four-byte System Backplane Bandwidth Allocation Table 0 register is writable and contains a core-dependent value on device reset. This register is used for system bus time-slice arbitration.

Note: This register is not used on the BCM4402.

Table 13: System Backplane Bandwidth Allocation Table 0 Register Bit Field Descriptions

Field	Name	Description	Access	Default
31:16	LookupTable1 (L1)	Lookup Table 1 This field contains the time slice allocation table indexed by the system backplane CCNTR1 counter (see "System Backplane Broadcast Configuration Register (SBBCONFIG, offset 0xFC0)" on page 22).	R/W	–
15:0	LookupTable0 (L0)	Lookup Table 0 This field contains the time slice allocation table indexed by the system backplane CCNTR0 counter (see "System Backplane Broadcast Configuration Register (SBBCONFIG, offset 0xFC0)" on page 22).	R/W	–

The CCNTR0 and CCNTR1 counters increment continuously. This core is allocated a time slice if LookupTable0[CCNTR0] and LookupTable1[CCNTR1] are both set. Software must ensure that this condition is never true for more than one core simultaneously.

SYSTEM BACKPLANE INITIATOR CONFIGURATION LOW REGISTER (SBIMCONFIGLOW, OFFSET 0XFA8)

The four-byte SBIMCONFIGLOW register is writable and contains the value 0x000*n*0053 on device reset, where n is a unique value for each core on the system backplane. This register controls request timeout behavior for the core.

Table 14: System Backplane Initiator Configuration Low Register Bit Field Descriptions

Field	*Field*	*Description*	*Access*	*Default*
31:24	Reserved	–	R/W	–
23:16	ConnectionID (CI)	Connection ID This read-only field contains a core-specific unique identifier to identify the source of system backplane transactions. Software error recovery routines can compare logged connection IDs with this field in each core to determine the source of an errant transaction.	R/W	–
15:7	Reserved	–	R/W	–
6:4	RequestTimeout (RT)	Request TIme-out This field controls the request timeout interval for transfers initiated by this core. The encodings for this field are shown in Table 15. If a transfer has not been completed when a request timeout occurs, the request is discarded and the TimeoutErr field of the SBIMSTATE register (see "System Backplane Initiator State Register (SBIMSTATE, offset 0xF90)" on page 12) is set. The value of this field should be greater than or equal to the value of the ServiceTimeout field.	R/W	–
3	Reserved	–	R/W	–
2:0	ServiceTimeout (ST)	Service Time-out This field controls the service timeout interval for transfers initiated by this core. The encodings for this field are shown in Table 15. If a transfer has not been completed when a service timeout occurs, the request is re-issued as a high-priority transaction.	R/W	–

Table 15: SB Timeout Period

SBREQTIMEOUT [2:0]	*Timeout Period*
0 0 0	Timeout disabled
0 0 1	2^6 SiliconBackplane cycles
0 1 0	2^8 SiliconBackplane cycles
0 1 1	2^{10} SiliconBackplane cycles
1 0 0	2^{12} SiliconBackplane cycles
1 0 1	2^{14} SystemBackplane cycles
1 1 0	2^{16} SystemBackplane cycles
1 1 1	2^{18} SystemBackplane cycles

SYSTEM BACKPLANE INITIATOR CONFIGURATION HIGH REGISTER (SBIMCONFIGHIGH, OFFSET 0XFAC)

The four-byte SBIMCONFIGHIGH register is writable and contains the value 0x0 on device reset. This register controls error reporting for the core.

Table 16: System Backplane Initiator Configuration High Register Bit Field Descriptions

Field	*Field*	*Description*	*Access*	*Default*
31:8	Reserved	–	R/W	0
7:6	BusErrMode (BE)	Bus Error Mode This field controls the reporting of system backplane errors to this core. When this field is set to 0, system backplane errors are not reported to this core. System backplane errors are reflected to the core as errors when this field is set to 1, and are reflected as interrupts when this field is set to 2. The value 3 is reserved.	R/W	0
5:4	TimeoutErrMode (TE)	Time-out Error Mode When set to 1, system backplane time-outs on requests initiated by this core asserts the system backplane SBERROR signal. When set to 0, time-outs are not reported as system backplane errors. The values 2 and 3 are reserved. This field should be left set to 0.	R/W	0
3:2	InBandErrMode (IE)	InBand Error Mode When set to 1, in-band error responses to this core asserts the system backplane SBERROR signal. When set to 0, in-band errors are not reported as system backplane errors. The values 2 and 3 are reserved. This field should be left set to 0.	R/W	0
1:0	Reserved	–	R/W	0

See "Error Architecture" on page 6 for more information on error handling.

SYSTEM BACKPLANE ADDRESS MATCH 0 REGISTER (SBADMATCH0, OFFSET 0XFB0)

The four-byte SBADMATCH0 register is read-only and contains the value 0x0nnnn058 at device reset. This register defines the host interface address space for the core as the range 0xnnnnn000-0xnnnnnFFF. See "Address Space Map" on page 7 for the value of the AddressBase field for each core in the BCM440X.

Table 17: System Backplane Address Match 0 Register Bit Field Descriptions

Field	Name	Description	Access	Default
31:8	AddressBase (AB)	Address Base This field contains the base address for the core.	RO	–
7:3	AddressSize (AS)	Address Size This field contains the value 0xb, indicating that the address range size is 4KB.	RO	–
1:0	AddressType (AT)	Address Type This field contains 0, indicating that this is a positive decode address range.	RO	–

SYSTEM BACKPLANE TARGET CONFIGURATION LOW REGISTER (SBTMCONFIGLOW, OFFSET 0XFB8)

The 4-byte SBTMCONFIGLOW register is read-only and contains the value 0x30101 on device reset. This register controls clock and interrupt behavior for the target portion of the core.

Table 18: System Backplane Target Configuration Low Register Bit Field Descriptions

Field	*Name*	*Description*	*Access*	*Default*
31:26	Reserved	–	RO	0x0
25:24	InterruptMode (IM)	Interrupt Mode This field is set to 0 because the Interrupt Flag is not used.	RO	0x0
23:18	InterruptFlag (IF)	Interrupt Flag This field is set to 0 for all cores; the Interrupt Flag is not used.	RO	0x0
17:16	Reserved	–	RO	0x3
15:11	ClockOffset (CO)	Clock Offset This field is set to 0 for all cores.	RO	0x0
10:8	Reserved	–	RO	0x1
7:0	ClockDivide (CD)	Clock Divide This field is set to 1 for all cores.	RO	0x1

SYSTEM BACKPLANE TARGET CONFIGURATION HIGH REGISTER (SBTMCONFIGHIGH, OFFSET 0XFBC)

The four-byte SBTMCONFIGHIGH register is writable and contains the value 0x100 on device reset in most cores. In the SDRAM controller core, the reset value is 0x106. This register controls clock and interrupt behavior for the target portion of the core.

Table 19: System Backplane Target Configuration High Register Bit Field Descriptions

Field	*Name*	*Description*	*Access*	*Default*
31:12	Reserved	–	R/W	–
11:10	SBIntMode (IM)	System Backplane Interrupt Mode This read-only field is set to 0, indicating that interrupts from this core are not reported as system backplane errors.	R/W	–
9:8	SBErrorMode (EM)	System Backplane Error Mode This field controls the reporting of local errors to the system backplane SBERROR signal. When set to 0, local errors are not reported as system backplane errors. When set to 1, the SBERROR signal is asserted whenever the SError field of the SBTMSTATEHIGH register (see "System Backplane Target State High Register (SBTMSTATEHIGH, offset 0xF9C)" on page 15) is set to 1. The values 2 and 3 are reserved.	R/W	–
7:6	Reserved	–	R/W	–
5:4	StopMode (SM)	Stop Mode This read-only field is set to 0, indicating that BCM440X does not use the thread busy stop condition.	R/W	–
3:2	RetryMode (RM)	Retry Mode This read-only field is set to 0 for most cores, indicating that requests that receive retry responses from this core should be retried immediately by the initiating agent. It is set to 1 for the SDRAM controller core, showing that the SDRAM controller indicates via a flag when the request should be retried.	R/W	–
1:0	BusyMode (BM)	Busy Mode This read-only field is set to 0 in most cores, indicating that the busy flag mechanism is not used and a busy response is always returned when a request cannot be serviced. In the SDRAM controller core, this field is writable and the default value is 0x2, causing the SDRAM core to use a response flag to indicate when it is no longer busy.	R/W	–

SYSTEM BACKPLANE BROADCAST CONFIGURATION REGISTER (SBBCONFIG, OFFSET 0XFC0)

The four-byte System Backplane Broadcast Configuration register is writable and contains the value 0x00ff0002 at device reset. Because it affects the operation of the entire system backplane, this register can be written only with a broadcast access performed by writing 0xFC0 to the PCI core's BroadcastAddr register (see "Broadcast Address Register (BroadcastAddress, offset 0x50)" on page 173) and then writing the desired value to the PCI core's BroadcastDat**A** register (see "Broadcast Data Register (BroadcastData, offset 0x54)" on page 173). See "PCI Core" on page 165 for more information.

Table 20: System Backplane Broadcast Configuration Register Bit Field Descriptions

Field	*Name*	*Description*	*Access*	*Default*
31:24	Reserved	–	R/W	0x00
23:20	SBMaxCCNTR1 (C1)	System Backplane Maximum Cycle Counter 1 This read-only field contains the value 0xF, indicating that the CCNTR1 time slice arbitration counters cycles between 0 and 0xF. This field is 0 in cores that are not initiators.	R/W	0xF
19:16	SBMaxCCNTR0 (C0)	System Backplane Maximum Cycle Counter 0 This read-only field contains the value 0xF, indicating that the CCNTR0 time slice arbitration counters cycles between 0 and 0xF. This field is 0 in cores that are not initiators.	R/W	0xF
15:4	Reserved	–	R/W	0x000
3:0	SBLatency (LT)	System Backplane Latency The contents of this field specify the system backplane latency in system backplane clock cycles. The default value for this field is two. If it is changed, it must be change to a value between SBIDLOW.MinLatency and SBIDLOW.MaxLatency (see "System Backplane Identification Low Register (SBIDLOW, offset 0xFF8)" on page 25).	R/W	0x2

SYSTEM BACKPLANE BROADCAST STATE REGISTER (SBBSTATE, OFFSET 0XFC8)

The four-byte SB Broadcast State register is writable and contains the value 0x0 on device reset. The register contains fields that protect against updates to critical configuration register. Because it affects the operation of the entire system backplane, this register can be written only with a broadcast access performed by writing 0xFC8 to the PCI core's BroadcastAddr register (see "Broadcast Address Register (BroadcastAddress, offset 0x50)" on page 173) and then writing the desired value to the PCI core's BroadcastData register (see "Broadcast Data Register (BroadcastData, offset 0x54)" on page 173). See Section 6: "PCI Core" for more information.

Table 21: System Backplane Broadcast State Register Bit Field Descriptions

Field	*Name*	*Description*	*Access*	*Default*
31:2	Reserved	–	R/W	0
1	HoldRegDis1 (HD)	Hold Register Disable 1 When this field is set to 1, updates to the following registers are disabled: • SBIMCONFIGLOW (see "System Backplane Initiator Configuration Low Register (SBIMCONFIGLOW, offset 0xFA8)" on page 17) • SBIMCONFIGHIGH (see "System Backplane Initiator Configuration High Register (SBIMCONFIGHIGH, offset 0xFAC)" on page 18) • SBTMCONFIGLOW (see "System Backplane Target Configuration Low Register (SBTMCONFIGLOW, offset 0xFB8)" on page 20) • SBTMCONFIGHIGH (see "System Backplane Target Configuration High Register (SBTMCONFIGHIGH, offset 0xFBC)" on page 21) • SBBWA0 (see "System Backplane Bandwidth Allocation Table 0 Register (SBBWA0, offset 0xFA0)" on page 16) All address match registers other than SBADMATCH0 (see "System Backplane Address Match 0 Register (SBADMATCH0, offset 0xFB0)" on page 19) are also disabled. Broadcast writes to the SBBCONFIG register (see "System Backplane Broadcast Configuration Register (SBBCONFIG, offset 0xFC0)" on page 22) are not affected by this field.	R/W	0
0	StRegDis0 (SD)	State Register Disable 0 When this field is set to 1, updates to the following registers are disabled: • SBIMSTATELOW • SBIMSTATEHIGH • SBTMSTATELOW (see "System Backplane Target State Low Register (SBTMSTATELOW, offset 0xF98)" on page 14) • SBTMSTATEHIGH (see "System Backplane Target State High Register (SBTMSTATEHIGH, offset 0xF9C)" on page 15) • SBIPSFLAG (see "System Backplane Initiator OCP Slave Flag Control Register (SBIPSFLAG, offset 0xF08)" on page 9) Broadcast writes to the SBBSTATE register (see "System Backplane Broadcast State Register (SBBSTATE, offset 0xFC8)" on page 23) are not affected by this field.	R/W	0

System Backplane Activate Configuration Register (SBACTCNFG, offset 0xFD8)

The System Backplane Activate Configuration register supports the simultaneous updating of buffered configuration registers in all cores. To perform this update, software writes the value 0xFD8 to the PCI core's BroadcastAddr register (see "Broadcast Address Register (BroadcastAddress, offset 0x50)" on page 173) and then writes any value to the PCI core's BroadcastData register (see "Broadcast Data Register (BroadcastData, offset 0x54)" on page 173). The data sent with the broadcast write is ignored.

System Backplane Flag Status Register (SBFLAGST, offset 0xFE8)

The System Backplane Flag Status register is read-only and contains the current value of the system backplane flags. Bit 7 of this register is set in the Ethernet MAC cores to indicate that the core's associated MII pins are not accessible and cannot be used.

SYSTEM BACKPLANE IDENTIFICATION LOW REGISTER (SBIDLOW, OFFSET 0XFF8)

The four-byte SBIDLOW register is read-only and contains a core-dependent value on device reset. This section describes the standard BCM440X configuration; any differences from the standard are described in the OCP section of each core's documentation.

Table 22: System Backplane Identification Low Register Bit Field Descriptions

Field	Name	Description	Access	Default
31:28	Reserved	–	RO	–
27:24	InitPorts (IP)	Initiator Ports This field indicates the highest numbered initiator port in the core. It is normally set to 0.	RO	–
23:20	TargetPorts (TP)	Target Ports This field indicates the highest numbered target port in the core. It is normally set to 0.	RO	–
19:18	CycleCounterWidth (CC)	Cycle Counter Width This field is set to 1, indicating that the cycle counter width used by time slice arbitration is 8 bits. This field is 0 in cores that are not initiators.	RO	–
17	NoTarget (NT)	No Target This field is set to 1 in cores that do not contain target agents.	RO	–
16	FirstInitiator (FI)	First Initiator This field is set to 1 in one of the cores, indicating that it owns the system backplane arbitration token after a reset, and is set to 0 in all other cores.	RO	–
15:12	MaxLatency (MX)	Maximum Latency This field is set to 4, indicating that the maximum system backplane latency is four cycles.	RO	–
11:8	MinLatency (MN)	Minimum Latency This field is set to 2, indicating that the minimum system backplane latency is two cycles.	RO	–
7	Initiator (IT)	Initiator This field is set to 1 for all cores that initiate system backplane transactions and is cleared to 0 for cores that do not.	RO	–
6	Synch (SC)	Synchronous The field is set to 1 for all cores, indicating that the cores are synchronous.	RO	–
5:3	AddressRanges (AR)	Address Ranges The contents of this field indicate the number of address ranges supported by this core. The value of this field is core-dependent. See "Address Space Map" on page 7 for more information.	RO	–
2:0	ConfigSpace (CS)	Configuration Space This field is set to 4 for all cores, indicating that the configuration registers range is the last 256 bytes of the core's host interface address space.	RO	–

SYSTEM BACKPLANE IDENTIFICATION HIGH REGISTER (SBIDHIGH, OFFSET 0XFFC)

The four-byte SBIDHIGH register is read-only and contains a core-dependent value used to identify the core.

Table 23: System Backplane Identification High Register Bit Field Descriptions

Field	Name	Description	Access	Default
31:16	Vendor (VN)	Vendor This field contains the vendor ID for each core. The Broadcom vendor ID is 0x4243.	RO	–
15:4	Core (CR)	Core This field contains the core type. See Table 24 for a list of Broadcom OCP cores.	RO	See Table 24.
3:0	Revision (RV)	Revision This field contains the revision number of the core.	RO	–

Table 24: OCP Cores

Value	Core
0x801	InsideLine
0x803	SDRAM controller
0x804	PCI
0x805	MIPS
0x806	Ethernet MAC
0x807	Codec
0x808	USB
0x80D	PCMCIA
0x811	External
0x812	802.11 MAC

Section 3: DMA

The BCM440X implements a distributed DMA architecture where each core can include a common implementation of one or more DMA processors. These DMA processors are used to transfer data between a core and memory. Data transfers between cores are not supported.

Each DMA processor consists of a transmit channel and a receive channel. The transmit channel transfers data from memory to a core and the receive channel transfers data from a core to memory.

DMA transfers to and from the PCI bus are described in Section 6: "PCI Core".

Register Descriptions

The DMA registers include the following:

TRANSMIT CHANNEL CONTROL REGISTER (XMTCONTROL, OFFSET 0X200)

The four-byte Transmit Channel Control register is both readable and writable and contains the value 0x0 at device reset.

Table 25: Transmit Channel Control Register Bit Field Descriptions

Field	Name	Description	Access	Default
31:5	Reserved	–	R/W	0
4	Flush (FL)	Flush When this field is set to 1, the descriptor processor indicates that the core should abandon processing of the current transmit frame if possible. This field must be cleared to 0 after the XmtState field of the XmtStatus register (see "Transmit Channel Status Register (XmtStatus, offset 0x20C)" on page 31) indicates that the transmit channel is idle. Not all cores support this operation. See the documentation for individual cores for more information.	R/W	0
3	FairPriority (FP)	Fair Priority When this field is set to 1, the descriptor processor uses round-robin arbitration between the transmit and receive channels. When this field is cleared to 0, receive channel transfers have priority over transmit channel transfers.	R/W	0
	Reserved	BCM4401 B0 only	R/W	0
2	LoopbackEn (LE)	DMA Loopback Enable When set to 1, this field causes all transmit channel data to be looped back to the receive channel. When this field is cleared to 0, transmit data is sent to the core. Not all cores support DMA loopback.	R/W	0
1	SuspEn (SE)	Transmit Suspend Request When set to 1, this field causes activity on the transmit channel to be suspended after the end of the current frame. If the channel is currently idle, the suspension takes effect immediately. After it is suspended, the channel remains suspended until this field is cleared to 0.	R/W	0
0	XmtEn (XE)	Transmit Enable This field enables activity on the transmit channel. When the enable bit is set to 1, the DMA engine enters an idle wait state until the software indicates that valid descriptors are in the table by writing to the XmtPtr register (see "Transmit Descriptor Table Pointer Register (XmtPtr, offset 0x208)" on page 30). While this field is cleared to 0, all transmit states for this processor—including any pending error interrupts—are reset. When this field is cleared to 0, the channel is not disabled until pending bus operations have been completed.	R/W	0

TRANSMIT DESCRIPTOR TABLE ADDRESS REGISTER (XMTADDR, OFFSET 0X204)

The four-byte Transmit Descriptor Table Address register is both readable and writable and is initialized to 0x0 at device reset. The register contains the memory space address of first byte of the transmit descriptor table (see "Transmit Descriptors" on page 40).

The lower 12 bits of the address of the transmit descriptor table are fixed at 0, ensuring that the descriptor table is aligned on a 4KB boundary.

Table 26: Transmit Descriptor Table Address Register Bit Field Descriptions

Field	*Name*	*Description*	*Access*	*Default*
31:12	BaseAddr (BA)	Base Address This field contains the upper 20 bits of the address of the transmit descriptor table for this channel.	R/W	0
11:0	Reserved	–	R/W	0

TRANSMIT DESCRIPTOR TABLE POINTER REGISTER (XMTPTR, OFFSET 0X208)

The four-byte Transmit Descriptor Table Pointer register is both readable and writable and is initialized to 0x0 at device reset.

Table 27: Transmit Descriptor Table Pointer Register Bit Field Descriptions

Field	Name	Description	Access	Default
31:12	Reserved	–	R/W	0
11:0	LastDscr (LD)	Last Valid Descriptor Device drivers should write this register with the byte offset of the descriptor following the last valid descriptor that is available for data transfer. If the last valid descriptor has the EoT field set, this field should be set to 0. The value of the LastDscr field cannot be written with the value currently in the CurrDscr field. This field is never updated by the descriptor processor.	R/W	0

When the value in the LastDscr field is the same as the value in the CurrDscr field in the XmtStatus register (see "Transmit Channel Status Register (XmtStatus, offset 0x20C)" on page 31), the descriptor table is empty and the transmit processor is idle. When LastDscr plus one descriptor size equals CurrDscr, the descriptor table (see "Transmit Descriptors" on page 40) is full.

If LastDscr addresses a descriptor at a higher offset than a descriptor with the EoT field set, the descriptor processor continuously fetches descriptors between 0 and that descriptor in a round-robin fashion.

After host software updates LastDscr to designate a descriptor as available for hardware use, it cannot change LastDscr to remove that descriptor from hardware use without first resetting the channel.

The value of this register is fixed at 0x0 while the channel is disabled.

TRANSMIT CHANNEL STATUS REGISTER (XMTSTATUS, OFFSET 0X20C)

The four-byte Transmit Channel Status register is read only and contains the value 0x0 at device reset.

Table 28: Transmit Channel Status Register Bit Field Descriptions

Field	*Name*	*Description*	*Access*	*Default*
31:20	Reserved	–	RO	0
19:16	XmtErr (XE)	Transmit Error This field indicates that an error occurred processing the descriptor entry addressed by CurrDscr. The errors that can occur are shown in the Transmit Channel Errors (see Table 30 on page 32). To reset the error status, the driver should clear the XmtEn field in the XmtControl register (see "Transmit Channel Control Register (XmtControl, offset 0x200)" on page 28), read the XmtStatus register until the XmtState field indicates that the channel is disabled, and then reassert the XmtEn field.	RO	0
15:12	XmtState (XS)	Transmit State This field indicates the current state of the transmit processor. The current states are shown in the Transmit Channel Status (see Table 29 on page 32). A channel is disabled when the XmtEn field of the XmtControl register (see "Transmit Channel Control Register (XmtControl, offset 0x200)" on page 28) is clear. An active channel is currently processing descriptors. An enabled channel with no descriptors ready to process is in the idle wait state. A channel is stopped when an error occurs and suspend-pending between the time the SuspEn field of the XmtControl register is set to 1 and the time the channel becomes idle by completing or flushing the current frame. When the descriptor processor channel is in the stopped state, the CurrDscr field still points to the first descriptor of the frame that was being processed when the error occurred. The host software must examine every descriptor starting with CurrDscr to determine which one caused the error. In test mode, CurrDscr points to the descriptor that was being processed when the error occurred. The stopped state can be exited only by clearing the XmtEn field of the XmtControl register	RO	0
11:0	CurrDscr (CD)	Current Descriptor Pointer This field contains the byte offset of the descriptor entry that the chip is currently processing. This field is updated when the entire frame has been transferred to the core.	RO	0

Table 29: Transmit Channel Status

Condition	*Value*
Disabled	0x0
Active	0x1
Idle Wait	0x2
Stopped	0x3
Suspend Pending	0x4
Reserved	0x5-0xF

Table 30: Transmit Channel Errors

Condition	*Value*
No Error	0x0
Descriptor Protocol Error	0x1
Data FIFO Underrun	0x2
Data Transfer Error	0x3
Descriptor Read Error	0x4
Reserved	0x5-0xF

The value of this register is fixed at 0x0 while the channel is disabled.

RECEIVE CHANNEL CONTROL REGISTER (RCVCONTROL, OFFSET 0X210)

The four-byte Receive Channel Control register is both readable and writable and contains the value 0x40 at device reset.

Table 31: Receive Channel Control Register Bit Field Descriptions

Field	*Name*	*Description*	*Access*	*Default*
31:9	Reserved	–	R/W	–
8	FIFOMode (FM)	FIFO Mode When this field is set to 1, receive DMA is suppressed and host software must retrieve received data and status information using the Direct FIFO access registers. When cleared to 0, received data and status are transferred to memory by the descriptor processor. This field is not implemented in all cores.	R/W	–
7:1	RcvOffset (RO)	Receive Frame Offset This field specifies the size (in bytes) of the status region at the beginning of the receive frame buffer. This value must be large enough to contain the core's per-frame status information. The size of this information is core-dependent; see the documentation for the receive frame data header for each core. Device drivers can reserve additional space for their own use. All receive buffers must be larger than this offset or a descriptor protocol error results.	R/W	–
0	RcvEn (RE)	Receive Enable This field enables activity on the receive channel. When the enable bit is set to 1, the DMA engine enters an idle wait state until a frame is received at a priority assigned to this channel. Prior to the reception of a frame, the host should indicate that valid descriptors are in the table by writing to the RcvPtr register (see "Receive Descriptor Table Pointer Register (RcvPtr, offset 0x218)" on page 35). While this field is cleared to 0, all receive states for this processor—including any pending error interrupts—are reset. When this field is cleared to 0, the channel is not disabled until pending bus operations have been completed.	R/W	–

RECEIVE DESCRIPTOR TABLE ADDRESS REGISTER (RCVADDR, OFFSET 0X214)

The four-byte Receive Descriptor Table Address register is both readable and writable and is initialized to 0x0 at device reset. The register contains the memory space address of the first byte of the receive descriptor table (see "Receive Descriptors" on page 40).

The lower 12 bits of the address of the receive descriptor table are fixed at 0, ensuring that the descriptor table is aligned on a 4 KB boundary.

Table 32: Receive Descriptor Table Address Register Bit Field Descriptions

Field	*Name*	*Description*	*Access*	*Default*
31:12	BaseAddr (BA)	Base Address This field contains the upper 20 bits of the address of the receive descriptor table for this channel.	R/W	0
11:0	Reserved	–	R/W	0

RECEIVE DESCRIPTOR TABLE POINTER REGISTER (RCVPTR, OFFSET 0X218)

The four-byte Receive Descriptor Table Pointer register is both readable and writable and is initialized to 0x0 at device reset.

Table 33: Receive Descriptor Table Pointer Register Bit Field Descriptions

Field	Name	Description	Access	Default
31:12	Reserved	–	R/W	0
11:0	LastDscr (LD)	Last Valid Descriptor Device drivers should write this register with the byte offset of the descriptor following the last valid descriptor that is available for data transfer. If the last valid descriptor has the EoT field set, this field should be set to 0. The value of the LastDscr field cannot be written with the value currently in the CurrDscr field. This field is never updated by the descriptor processor.	R/W	0

When the value of LastDscr is the same as the value CurrDscr in the RcvStatus register (see "Receive Channel Status Register (RcvStatus, offset 0x21C)" on page 36), the descriptor table is empty. When LastDscr plus one descriptor size equals CurrDscr, then the descriptor table (see "Receive Descriptors" on page 40) is full.

If LastDscr addresses a descriptor at a higher offset than a descriptor with the EoT field set, the descriptor processor continuously fetches descriptors between 0 and that descriptor in a round-robin fashion.

After host software updates LastDscr to designate a descriptor as available for hardware use, it cannot change LastDscr to remove that descriptor from hardware use without first resetting the channel.

The value of this register is fixed at 0x0 while the channel is disabled.

RECEIVE CHANNEL STATUS REGISTER (RCVSTATUS, OFFSET 0X21C)

The four-byte, read-only Receive Channel Status register contains the value 0x0 at device reset.

Table 34: Receive Channel Status Register Bit Field Descriptions

Field	Name	Description	Access	Default
31:20	Reserved	–	RO	0
19:16	RcvErr (RE)	Receive Error This field indicates that an error has occurred processing the descriptor entry addressed by the CurrDscr field. The errors that can occur shown in the Receive Channel Errors (see Table 36 on page 37). To reset the error status, the driver should clear the RcvEn field in the RcvControl register (see "Receive Channel Control Register (RcvControl, offset 0x210)" on page 33), read the RcvStatus register until the RcvState field indicates that the channel is disabled, and then reassert the RcvEn field.	RO	0
15:12	RcvState (RS)	Receive State This field indicates the current state of receive processing for this channel. The states are shown in the Receive Channel State (see Table 35 on page 37). A channel is disabled when the RcvEn field of the RcvControl register (see "Receive Channel Control Register (RcvControl, offset 0x210)" on page 33) is clear. An active channel is currently processing descriptors. An enabled channel awaiting an incoming frame is in the idle wait state. A channel is stopped when an error occurs. When the descriptor processor channel is in the stopped state, the CurrDscr field still points to the first descriptor of the frame that was being processed when the error occurred. The host software must examine every descriptor starting with CurrDscr to determine which one caused the error. In test mode, CurrDscr points to the descriptor that was being processed when the error occurred. The stopped state can be exited only by clearing the RcvEn field of the RcvControl register	RO	0
11:0	CurrDscr (CD)	Current Descriptor Pointer This field contains the byte offset of the descriptor entry that the chip is currently processing. This field is updated when the entire frame, including status information, has been transferred to memory.	RO	0

Table 35: Receive Channel State

Condition	*Value*
Disabled	0x0
Active	0x1
Idle Wait	0x2
Stopped	0x3
Reserved	0x4-0xF

Table 36: Receive Channel Errors

Condition	*Value*
No Error	0x0
Descriptor Protocol Error	0x1
Receive Data FIFO Overflow	0x2
Data Transfer Error	0x3
Descriptor Read Error	0x4
Reserved	0x5-0xF

The value of this register is fixed at 0x0 while the channel is disabled.

PROGRAMMER'S MODEL

Each DMA Processor contains a transmit channel and a receive channel that use a descriptor table (see "Descriptor Tables" on page 40) and a set of control and status registers (see "Register Descriptions" on page 27) to transfer data to or from host memory. Each channel also reports transfer completions (see "Receive Frame Status Information" on page 42) and error conditions (see "Error Conditions" on page 43) through its core's interrupt status register.

Each channel traverses a descriptor table and processes individual descriptors. Descriptors are used to pass information to the DMA processors. The processors are never modified by hardware. Status registers are used to return information from the DMA processors. Each descriptor points to a single data buffer. Important properties of data buffers are:

- The DMA processor does not impose alignment restrictions on either the start address or the length of data buffers. Individual cores can impose such restrictions. Those restrictions, if any, are described in the Descriptor Processor section of each core's documentation.
- Data buffers can have an arbitrary length up to the maximum size of 4 KB.
- The last transmit descriptor for a frame on the transmit channel must have a non-zero length. Zero-length buffers are permitted in all other cases.

DESCRIPTOR STRUCTURE

Transmit and Receive Descriptor Table entries have similar layouts (see Table 37). However, some of the individual fields within the descriptor are treated differently for transmit (see "Transmit Descriptors" on page 40) and receive (see "Receive Descriptors" on page 40).

A device driver uses the DataBufPtr and BufCount fields to associate the descriptor with a data buffer. The additional flag fields allow software to provide frame-based control information to the core.

Table 37: Transmit and Receive Descriptor Table Entries

Field	*Name*	*Description*
63:32	DATABUFPTR (DB)	Data Buffer Pointer This field contains the memory address of the first byte of the data buffer.
31	SOF (SF)	Start of Frame When set to 1, this field indicates that the current descriptor is the first one in the current frame. This field is ignored in a receive descriptor table entry.
30	EOF (EF)	End of Frame When set to 1, this field indicates that the current descriptor is the last one in the current frame. This field is ignored in a receive descriptor table entry.

Table 37: Transmit and Receive Descriptor Table Entries (Cont.)

Field	Name	Description
29	IOC (IC)	Interrupt on Completion When this field is set to 1, the descriptor processor channel generates an interrupt when the transfer of data to or from the buffer associated with this descriptor completes. When this field is cleared to 0, the descriptor processor channel does not generate an interrupt when the transfer completes. In transmit descriptors, this field is ignored unless the EOF field is also set. For receive descriptors, this field should be set only when transferring unframed data. When transferring framed data, the behavior of this field for receive descriptors is undefined; the lazy receive interrupt mechanism should be used instead. See the description of the IntRecvLazy register in documentation for individual cores: • Section 4: "Codec Core (BCM4402 only)": not applicable • Section 5: "Ethernet 10/100 Core": see "Interrupt Receive Lazy Time-out Register (IntRecvLazy, offset 0x100)" on page 91 • Section 6: "PCI Core": not applicable Some cores do not support completion interrupts; see the Interrupts descriptions in each core for more information: • Section 4: "Codec Core (BCM4402 only)": see "Interrupts" on page 66 • Section 5: "Ethernet 10/100 Core": see "Interrupts" on page 161 • Section 6: "PCI Core": see "Interrupts" on page 166
28	EOT (ET)	End of Descriptor Table When set to 1, this field indicates that the current descriptor is the last entry in the table. The next descriptor is fetched from the address in the: • RcvAddr register (see "Receive Descriptor Table Address Register (RcvAddr, offset 0x214)" on page 34) for receive descriptors • XmtAddr register (see "Transmit Descriptor Table Address Register (XmtAddr, offset 0x204)" on page 29) for transmit descriptors
27:20	FLAGS (FL)	Descriptor Core Flags This field is available for use by individual cores for core-specific purposes. Documentation for these flags can be found in the chapters describing individual cores that use them. • Section 4: "Codec Core (BCM4402 only)": see "Descriptor Core Flags" on page 66 • Section 5: "Ethernet 10/100 Core": not applicable • Section 6: "PCI Core": not applicable
12:0	BUFCOUNT (BC)	Buffer Byte Count This field is the length, in bytes, of the data buffer associated with this descriptor. A descriptor with a BufCount value greater than 0x1000 causes a descriptor protocol error.

DESCRIPTOR TABLES

A descriptor table is a linear array of descriptor entries (see "Descriptor Structure" on page 38) containing up to 512 entries. A descriptor table must be aligned on a 4 KB boundary. The limit of 512 entries ensures that the entire table is contained within a single 4 KB page. The actual size of a descriptor table is denoted by writing an end of table marker (EOT) in the last descriptor of the table. A sequence of 512 descriptor entries without an EOT marker is reported as a Descriptor Protocol error.

Transmit Descriptors

Multiple transmit descriptors are necessary if the data to be transferred is either fragmented or longer than 4 KB.

The host prepares all the descriptors that make up a frame by setting the DataBufPtr and BufCount fields to describe the buffers associated with the frame. The first descriptor must have the SOF field set and the last descriptor must have the EOF field set. The data buffer pointer in a descriptor with SOF set should point to the first byte of the data to be transferred. A completion interrupt is sent if the IOC field of the last descriptor is set. A descriptor protocol error occurs if the SOF and EOF fields are not set in the proper descriptors, or if the EOF field is set in a descriptor with a buffer size of 0.

After the descriptors have been initialized, software posts them by updating the LastDscr field of the XmtPtr register with the descriptor table byte offset of the descriptor following the last initialized descriptor. The hardware processes the descriptor table in a circular manner so if the last descriptor used is the last descriptor in the table, the new value of the LastDscr field should be 0.

After the descriptors have been queued on an enabled channel, the descriptor processor begins reading descriptors sequentially from the table, starting with the entry addressed by the CurrDscr field in the XmtStatus register (see "Transmit Channel Status Register (XmtStatus, offset 0x20C)" on page 31). As each descriptor is read, the buffer referenced by the DataBufPtr field in the descriptor is fetched, and the data is transferred to the core. After the core reports that the entire frame has been processed, CurrDscr is updated to point to the descriptor following the last descriptor used for this frame.

If the DMA processor does not supply the entire frame data before the core requires it for transmission, a transmit FIFO underrun interrupt occurs. This is an error, and all transmit processing stops until the channel is reset.

Receive Descriptors

The receive descriptors are handles to buffers that the host has setup to receive frame data. Received frames can span multiple descriptors if the incoming frame is larger than the buffer associated with the current descriptor. The host initializes descriptors by writing the DataBufPtr and BufCount fields with valid information for an empty buffer. Software makes the new descriptors available for use by updating the LastDscr field of the XmtPtr register (see "Receive Descriptor Table Pointer Register (RcvPtr, offset 0x218)" on page 35) to point to the entry after the new last valid descriptor.

The offset of the next descriptor to be used for receive data is found in the CurrDscr field of the RcvStatus register (see "Receive Channel Status Register (RcvStatus, offset 0x21C)" on page 36). The first RcvControl.RcvOffset bytes of this buffer are reserved for core-dependent status information. When a core has data available, it transfers it into one or more buffers, starting immediately after the status region of the first descriptor buffer. When the last buffer is written, status information, if any, is written back to the beginning of the first descriptor buffer. After the received frame and status information have been successfully transferred to memory, the CurrDscr field of the RcvStatus register is updated to point to the next unused descriptor in the table.

When the CurrDscr field points to the same descriptor as the LastDscr field in the XmtPtr register, the descriptor table is empty. If the receive descriptor table is empty and a core has valid data to transfer, a receive descriptor underflow interrupt occurs. This condition is not an error; when software supplies additional descriptors, processing resumes.

If the core's receive data FIFO becomes full before additional descriptors are supplied, a receive FIFO overrun interrupt occurs. This is an error, and all receive processing stops until the channel is reset.

For cores that receive unframed data, no receive status information is supplied, the RcvOffset field in the RcvControl register (see "Receive Channel Control Register (RcvControl, offset 0x210)" on page 33) is ignored, and the CurrDescr field in the RcvStatus register is updated each time a descriptor buffer is filled.

Descriptor Processor Channel Initialization

In order to avoid race conditions that can lead to errors and lost frames, host device drivers should obey a strict order for initializing a channel. For a receive channel, perform the following steps:

1. Set the RCVEN field in the RcvControl register (see "Receive Channel Control Register (RcvControl, offset 0x210)" on page 33).
2. Write the address of the receive descriptor table to the RcvAddr register (see "Receive Descriptor Table Address Register (RcvAddr, offset 0x214)" on page 34).
3. Initialize enough descriptors to contain an entire frame and update the RcvPtr register (see "Receive Descriptor Table Pointer Register (RcvPtr, offset 0x218)" on page 35) to point to the descriptor following the last initialized descriptor.

For a transmit channel, initialization is easier because software controls when transmit data is available. Before sending a frame, perform these two steps:

1. Set the XMTEN field in the XmtControl register (see "Transmit Channel Control Register (XmtControl, offset 0x200)" on page 28).
2. Write the address of the transmit descriptor table to the XmtAddr register (see "Transmit Descriptor Table Address Register (XmtAddr, offset 0x204)" on page 29).

Descriptor Processor Channel Reset

Descriptor processor channel state can be reset by clearing and then setting the enable fields in the channel control registers. These enable fields are XmtEn in the XmtControl register (see "Transmit Channel Control Register (XmtControl, offset 0x200)" on page 28) for the transmit channel and RcvEn in the RcvControl register (see "Receive Channel Control Register (RcvControl, offset 0x210)" on page 33) for the receive channel. This action clears all error conditions and interrupts associated with the channel, and also clears the current descriptor pointers to 0.

Host Bus Arbitration Between Receive and Transmit

The descriptor processor normally grants precedence to copying receive data to memory over copying transmit data from memory. If the FairPriority field of the XmtControl register (see "Transmit Channel Control Register (XmtControl, offset 0x200)" on page 28) is set, the processor uses round-robin arbitration between the transmit and receive channels.

RECEIVE FRAME STATUS INFORMATION

The receive descriptor process reserves a region at the beginning of buffer associated with the first descriptor of every frame for core-dependent status information. The size of this region is set to the value of the RcvOffset field of the RcvControl register (see "Receive Channel Control Register (RcvControl, offset 0x210)" on page 33). Status information is transferred to this region after all of the frame data has been transferred. Status information is not provided for cores that transfer unframed data.

The format of the first four bytes of the status information, shown in Table 38, is common across all cores that provide status information. The remaining fields are core dependent.

- Section 4: "Codec Core (BCM4402 only)":
- Section 5: "Ethernet 10/100 Core":
- Section 6: "PCI Core":

Table 38: Receive Frame Status Fields

Field	*Name*	*Description*
27:24	DescrCnt (DC)	Descriptor Count This field contains one less than the number of DMA descriptors used to transfer this frame into memory.
23:16	Reserved	Reserved
15:0	FrameLen (CD)	Frame Length The field contains the number of bytes in the received frame.

If a receive channel descriptor error occurs before the entire frame has been received, the status information is not written to host memory. In fact, the contents of the first status region are not altered when an error occurs. Some portion of the data that was received before the error occurred can be written to memory before the channel halts. In this case, the channel is stopped without updating the CurrDescr field of the RcvStatus register (see "Receive Channel Status Register (RcvStatus, offset 0x21C)" on page 36) to show that this partially processed frame was received.

ERROR CONDITIONS

The descriptor processor reports transfer completions and errors to its core. These conditions are visible as interrupts in the IntStatus register of any core containing a DMA processor. See the description of the IntStatus and IntMask registers in individual cores (see Table 39) for the exact location of these fields.

Table 39: IntStatus and IntMask Registers

Core	*IntStatus Register*	*IntMask Register*
Section 4: "Codec Core (BCM4402 only)"	See "Interrupt Status Register (IntStatus, offset 0x20)" on page 53.	See "Interrupt Mask Register (IntMask, offset 0x24)" on page 55.
Section 5: "Ethernet 10/100 Core"	See "Interrupt Status Register (IntStatus, offset 0x20)" on page 77.	See "Interrupt Mask Register (IntMask, offset 0x24)" on page 80.
Section 6: "PCI Core"	See "Interrupt Status Register (IntStatus, offset 0x20)" on page 170.	See "Interrupt Mask Register (IntMask, offset 0x24)" on page 171.

The descriptor processor notifies the core when transmit and receive completions occur. The mechanism for reporting and clearing these interrupts is core-specific. Resetting the channel does not clear completion interrupts.

The descriptor process reports three error conditions to the core: descriptor read error, data transfer error, and descriptor protocol error. An additional warning condition, receive descriptor underflow, is also reported. The core reports receive FIFO overflow and transmit FIFO underflow error conditions to the descriptor processor. Error conditions cause the channel to stop processing descriptors; the channel must be reset to continue operations.

DESCRIPTOR READ ERROR

This condition is reported when an error occurs while reading a descriptor from host memory. The likely cause of such an error is an invalid address in the channel's table address pointer register.

DATA TRANSFER ERROR

This condition is reported when an error occurs while transferring data to or from memory. The likely cause of such an error is an invalid buffer address in a descriptor. This interrupt is cleared by resetting the channel that caused the error.

DESCRIPTOR PROTOCOL ERROR

This condition is reported when a descriptor processor encounters a programming error while interpreting a descriptor. Descriptor protocol errors include receive descriptor buffers with invalid lengths, descriptors with reserved bits set, and missing start-of-frame, end-of-frame, and end-of-table indicators. This interrupt is cleared by resetting the channel that caused the error.

RECEIVE DESCRIPTOR UNDERFLOW

This condition is reported when a channel cannot process an incoming frame because no descriptors are available. This interrupt is cleared by supplying descriptors to the receive channel. If no descriptors are provided before the core's receive FIFO becomes full, a receive FIFO overflow error and interrupt also occurs.

Section 4: Codec Core (BCM4402 only)

Note: This section is not applicable to the BCM4401.

CODEC REGISTERS

Table 40: Codec Host Interface Register Address Spaces

Address Space	*Offset Range*
Device and Power Control	0x00-0x1F
Interrupt Control	0x20-0x2F
Codec Control	0x30-0x3F
Reserved	0x40-0xCF
Audio/IR	0xD0-0xFF
Reserved	0x100-0x1FF
Descriptor Processor	0x200-0x2FF
Reserved	0x300-0xEFF
System Backplane Configuration Space	0xF00-0xFFF

Table 41: Codec Host Interface Registers

Register	*Offset*	*Size*	*Type*	*Reset Value*	*Page*
Device Control – DevControl	0x00	4	R/W	0x181040	45
BIST Status - BISTStatus	0x0C	4	R	0x0	47
Interrupt Status – IntStatus	0x20	4	R/W	0x0	48
Interrupt Mask – IntMask	0x24	4	R/W	0x0	50
General Purpose Timer – GPTimer	0x28	4	R/W	0x0	51
Codec Control Command – CodecCtlCommand	0x38	4	W	undefined	52
Codec Control Status – CodecCtlStatus	0x3C	4	R	0x0	53

CODEC DEVICE AND POWER CONTROL REGISTERS

The codec device and power control registers are as follows:

- "Device Control Register (DevControl, offset 0x00)" on page 45
- "Built-In Self Test Register (BISTStatus, offset 0x0C)" on page 47

DEVICE CONTROL REGISTER (DEVCONTROL, OFFSET 0X00)

The four-byte Device Control Register is both readable and writable. On initial power-up, this register contains the value 0x181040.

Table 42: Device Control Register Bit Field Descriptions

Field	Name	Description	Access	Default
31:21	Reserved	–	R/W	–
20:16	MClk (MC)	Master Clock This field contains a five-bit value that specifies the frequency of the CODEC_MCLK output signal in terms of the backplane clock frequency. The CODEC_MCLK signal frequency is 1/(2 * (MCLK + 1)) times the backplane frequency.	R/W	–
15:14	Reserved	–	R/W	–
13	CtlMode (CM)	Control Mode When this field is set to 1, the CodecCtlCommand (see "Codec Control Command Register (CodecCtlCommand, offset 0x38)" on page 52) and CodecCtlStatus (see "Codec Control Status Register (CodecCtlStatus, offset 0x3C)" on page 53) registers access the Codec control registers. When this field is cleared to 0, the Codec control registers are accessed through the FIFO.	R/W	–
12	CodecReset (CR)	Codec Reset When set to 1, the device asserts the CODEC_RST_L signal to the Si3034 codec chip. When this field is cleared to 0, the CODEC_RST_L signal is not asserted. This field is set at power-up reset.	R/W	–
11	CodecIFMode (IF)	Codec IF Mode When this field is set to 1, the V.90 core communicates with the DAA_CHIP in pulsed mode. When cleared to 0, the V.90 core communicates with the DAA_CHIP in framed mode. See the documentation of the DAA_CHIP for more information.	R/W	–
10:7	Reserved	–	R/W	–
6	AudioReset (AR)	Audio Reset When this field is set to 1, the CODEC_RESET pin is asserted, resetting the audio interface. When cleared to 0, the CODEC_RESET pin is deasserted.	R/W	–
5	AudioMode (AM)	Audio Mode When both this field and the IRMode field are set to 1, the core operates in Audio Mode. When cleared to 0, the core operates in V.90 or IR mode. See "Audio/IR Registers" on page 54 for more information on the Audio interface.	R/W	–
4	IRMode (IM)	IR Mode When this field is set to 1, the core operates in IR Mode. When cleared to 0, the core operates in V.90 mode. See "Audio/IR Registers" on page 54 for more information on the IR interface.	R/W	–
3:1	Reserved	–	R/W	–

Table 42: Device Control Register Bit Field Descriptions (Cont.)

Field	Name	Description	Access	Default
0	OffHook (OH)	Off Hook The value in this field is copied to the CODEC_OFHK_L pin of the BCM440X to be sent to the external codec/DAA.	R/W	–

The codec core's external interface includes pins that are shared between an external DAA used in V.90 mode, an Audio interface, and an IR transceiver. The DAA interface cannot be used if either the Audio IR interfaces is in use. Table 43 shows which interfaces are active depending on the settings of the IRMode and AudioMode fields of the DevControl register.

Table 43: Active Interfaces

IM	AM	Active Interfaces
0	X	V.90 DAA
1	0	IR
1	1	IR in, Audio out

Mode-dependent changes to the external interface are detailed in Table 44.

Table 44: Mode-Dependent Changes

Pin Name	V.90	IR	Audio Out/IR In
CODEC_MCLK	CODEC_MCLK(O)	Not used	AUDIO_LRCK(O)
CODEC_SDO	CODEC_SDO(O)	IR_SDO(O)	AUDIO_SDO(O)
CODEC_SDI	CODEC_SDI(I)	IR_SDI(I)	IR_SDI(I)
CODEC_OFHK	CODEC_OFHK(O)	Not Used	AUDIO_SCLK(O)
CODEC_SCLK	CODEC_SCLK(I)	Not Used	AUDIO_CTRL_CLK(O)
CODEC_FSYNC	CODEC_FSYNC(I)	Not Used	AUDIO_CTRL_DATA(I/O)
CODEC_RGDT	CODEC_RGDT(I)	Not Used	AUDIO_MCLK(I)

BUILT-IN SELF TEST REGISTER (BISTSTATUS, OFFSET 0X0C)

The four-byte BISTStatus register is writable and contains the value 0x0 at device reset.

Table 45: Interrupt Status Register Bit Field Descriptions

Field	Name	Description	Access	Default
31:2	Reserved		R/W	0
1:0	BISTStatus	Power Management Event This field contains the result of the Built-In Self Test for all of The RAMs in the Codec core. If any bit is set to 1, the BIST failed for the corresponding RAM. See Table 46 for the mapping of RAMs to bits within this field. This field is valid only when the BISTDone field is set in the SBTMSTATEHIGH register (see "System Backplane Target State High Register (SBTMSTATEHIGH, offset 0xF9C)" on page 18).	R/W	0

Table 46: RAM-to-Bit Mapping

Bit	RAM
0	Receive DMA FIFO
1	Transmit DMA FIFO

CODEC INTERRUPT HANDLING REGISTERS

THe codec interrupt handling registers include the following:

INTERRUPT STATUS REGISTER (INTSTATUS, OFFSET 0X20)

The four-byte Interrupt Status register is both readable and writable and contains the status of all device interrupt sources. A bit set in this register indicates that an interrupt has been generated. The interrupt signal is not asserted to the host unless the corresponding enable bit in the IntMask register (see "Interrupt Mask Register (IntMask, offset 0x24)" on page 50) is set to 1. This register has the value 0x0 at device reset.

Table 47: Interrupt Status Register Bit Field Descriptions

Field	Name	Description	Access	Default
31:25	Reserved	–	R/W	0
24	XmtInt (XI)	Transmit Notification Interrupt This field contains a transmit notification interrupt bit for the channel. This field is set to 1 when a transmit completion interrupt is generated for the channel. Writing a 1 to this field clears both the field and the pending interrupt. Disabling the transmit channel does not clear this interrupt.	R/W	0
23:17	Reserved	–	R/W	0
16	RcvInt (RI)	Receive Notification Interrupt This field contains a receive notification interrupt bit for the channel. This field is set to 1 when a receive completion interrupt is generated for the channel. Writing a 1 this field clears both the field and the pending interrupt. Disabling the receive channel does not clear this interrupt.	R/W	0
15	XmtFIFOUf (XU)	Transmit FIFO Underflow This field is set to 1 when a channel cannot forward incoming data to the external Codec because the transmit FIFO is empty. When the descriptor processor is enabled, this field is cleared to 0 by resetting the transmit channel.	R/W	0
14	RcvFIFOOf (RO)	Receive FIFO Overflow This field is set to 1 when a channel cannot process incoming data because the receive FIFO is full. When the descriptor processor is enabled, this field is cleared to 0 by resetting the receive channel.	R/W	0
13	RcvDescUf (RU)	Receive Descriptor Underflow This field is set to 1 when a channel cannot process incoming data because no descriptors are available. This field is cleared to 0 by supplying descriptors to the receive channel. If no descriptors are provided before the receive FIFO becomes full, a receive FIFO overflow error and interrupt also occurs.	R/W	0

Table 47: Interrupt Status Register Bit Field Descriptions (Cont.)

Field	Name	Description	Access	Default
12	DescErr (DE)	Descriptor Protocol Error This field is set to 1 when a descriptor processor encounters a programming error while interpreting a descriptor. Descriptor protocol errors include receive descriptor buffers with invalid lengths, descriptors with reserved bits set, and missing start-of-frame, end-of-frame, and end-of-table indicators. This field is cleared to 0 by resetting the channel that caused the error.	R/W	0
11	PCIData (PD)	PCI Data Error This field is set to 1 when a PCI bus error occurs while transferring data to or from host memory. The PCI status register (see "Status Register (Status, offset 0x06)" on page 184) can be read to determine the cause of the error. This field is cleared to 0 by resetting the channel that caused the error.	R/W	0
10	PCIDesc (PC)	PCI Descriptor Error This field is set to 1 when a PCI bus error occurs while reading a descriptor from host memory. The PCI status register (see "Status Register (Status, offset 0x06)" on page 184) can be read to determine the cause of the error. This field is cleared to 0 by resetting the channel that caused the error.	R/W	0
9:8	Reserved	–	R/W	0
7	TimeOut (TO)	Timeout This field is set to 1 when a general purpose timer timeout interrupt is generated. Writing a 1 to this field clears both the field and the pending interrupt.	R/W	0
6	RingDetect (RD)	Ring Detect This field is set to 1 when the external Codec detects a ring on the line.	R/W	0
5	PME	Power Management Event This field is set to 1 when a power management event is generated. Writing a 1 to this field clears both the field and the pending interrupt.	R/W	0
4:0	Reserved	–	R/W	0

INTERRUPT MASK REGISTER (INTMASK, OFFSET 0X24)

The four-byte Interrupt Mask register is both readable and writable and selectively enables the delivery of interrupts to the host processor. The enables in this register do not affect the recording of interrupt conditions in the status register (see "Interrupt Status Register (IntStatus, offset 0x20)" on page 48); they affect the propagation of the interrupt to the PCI bus INT_L signal only. This register has a value of 0x0 at device reset.

Table 48: Interrupt Mask Register Bit Field Descriptions

Field	Name	Description	Access	Default
31:25	Reserved	–	R/W	0
24	XmtIntEn (XI)	Transmit Interrupt Enable When set to 1, this field enables interrupt delivery from the transmit channel of the descriptor processor.	R/W	0
23:17	Reserved	–	R/W	0
16	RcvIntEn (RI)	Receive Interrupt Enable When set to 1, this field enables interrupt delivery from the receive channel of the descriptor processor.	R/W	0
15	XmtUfEn (XU)	Transmit Underflow Enable When set to 1, this field enables interrupts that occur when a channel cannot process an outgoing frame because no more data are available in the transmit FIFO.	R/W	0
14	RcvOfEn (RO)	Receive Overflow Enable When set to 1, this field enables interrupts that occur when a channel cannot process an incoming frame because no space is available in the receive data or receive status FIFOs.	R/W	0
13	RcvUfEn (RU)	Receive Underflow Enable When set to 1, this field enables interrupts that occur when a channel cannot process an incoming frame because no descriptors are available.	R/W	0
12	DescErrEn (DE)	Descriptor Protocol Error Enable When set to 1, this field enables interrupts on descriptor programming errors.	R/W	0
11	PCIDataEn (PD)	PCI Data Error Enable When set to 1, this field enables interrupts on PCI bus errors that occur while transferring data to or from host memory.	R/W	0
10	PCIDescEn (PC)	PCI Descriptor Error Enable When set to 1, this field enables interrupts on PCI bus errors that occur while reading a descriptor from host memory.	R/W	0
9:8	Reserved	–	R/W	0
7	TimeOut (TO)	Timeout When set to 1, this field enables general purpose timer timeout interrupts.	R/W	0
6	RingDetect (RD)	Ring Detect When set to 1, this field enables ring detect interrupts.	R/W	0
5	PME	Power Management Event This field is set to 1 when a power management event is generated. Writing a 1 to this field clears both the field and the pending interrupt.	R/W	0
4:0	Reserved	–	R/W	0

GENERAL PURPOSE TIMER REGISTER (GPTIMER, OFFSET 0X28)

The four-byte General Purpose Timer register is a free-running 32-bit counter that decrements in PCI cycles. The counter is initialized when either the whole word or the upper 16 bits are written. This register is both readable and writable and has the value 0x0 at device reset.

Table 49: General Purpose Timer Register Bit Field Descriptions

Field	*Name*	*Description*	*Access*	*Default*
31:0	–	Writing to this register initializes the time-out counter to the value written. When the counter decrements to 0, a general purpose timer time-out interrupt is generated. The counter is then loaded with the last value written and continues to decrement. Reading the register returns the current value of the timer. Writing a 0 to this register disables the counter.	R/W	0

CODEC CONTROL AND STATUS REGISTERS

The codec control and status registers are as follows:

CODEC CONTROL COMMAND REGISTER (CODECCTLCOMMAND, OFFSET 0X38)

The two-byte, write-only Codec Control Command register is used to access the registers in the Si3034. This register is ignored when the CtlMode field in the DevControl register (see "Device Control Register (DevControl, offset 0x00)" on page 45) is cleared.

Table 50: Codec Control Command Register Bit Field Descriptions

Field	*Name*	*Description*	*Access*	*Default*
15:14	Reserved	–	WO	–
13	Read (RD)	Read This field is set to 1 to read a Codec register. This field is cleared to 0 to write a Codec register.	WO	–
12:8	Address (AD)	Address This field contains the Codec register internal address to be written or read. See the *Si3034 Data Sheet* for Codec address mapping.	WO	–
7:0	Data (DA)	Write Data This field contains data to be written to a Codec register.	WO	–

Note: After writing this register, the host software should check the Busy field of the CodecCtlStatus register (see "Codec Control Status Register (CodecCtlStatus, offset 0x3C)" on page 53) to verify that the register access has completed.

CODEC CONTROL STATUS REGISTER (CODECCTLSTATUS, OFFSET 0X3C)

The two-byte, read-only Codec Control Status register is used to access the registers in the Si3034. This register should not be accessed when the CtlMode field in the DevControl register (see "Device Control Register (DevControl, offset 0x00)" on page 45) is cleared. This register contains the value 0x0 at device reset.

Table 51: Codec Control Status Register Bit Field Descriptions

Field	*Name*	*Description*	*Access*	*Default*
15	Busy (BU)	Busy This field is set to 1 when an access initiated by a write to the CodecCtlCommand register (see "Codec Control Command Register (CodecCtlCommand, offset 0x38)" on page 52) fails because the Codec interface is busy. This field is cleared to 0 when an access is successful.	RO	–
14:8	Reserved	–	RO	–
7:0	Data (DA)	Read Data This field contains the Codec data read as a result of a write to the CodecCtlCommand register (see "Codec Control Command Register (CodecCtlCommand, offset 0x38)" on page 52).	RO	–

AUDIO/IR REGISTERS

The audio/IR registers are as follows:

IRXmtClkDivideCount Register (IRXmtClkDivideCount, offset 0xD0)

The four-byte IRXmtClkDivideCount register is writable and contains the value 0x2000 at device reset. This register contains a 16-bit counter that specifies the period of the IR transmit clock in terms of the number of backplane clock cycles per IR transmit clock.

The frequency of the IR transmit clock should be between 1.6 and 76 KHz. A backplane clock frequency of 100 MHz and the default value of 0x2000 gives a clock frequency of about 6.1 KHz.

Table 52: IRXmtClkDivideCount Register Bit Field Descriptions

Field	Name	Description	Access	Default
31:16	Reserved	–	R/W	–
15:0	–	–	R/W	0x2000

IRRcvClkDivideCount Register (IRRcvClkDivideCount, offset 0xD4)

The four-byte IRRcvClkDivideCount register is writable and contains the value 0x1000 at device reset. This register contains a 16-bit counter that specifies the period of the IR receive clock in terms of the number of 125 MHz clock cycles per IR receive clock.

The frequency of the IR transmit clock should be between 3 and 76 KHz. A backplane clock frequency of 100 MHz and the default value of 0x1000 gives a clock frequency of about 12.2 KHz.

Table 53: IRRcvClkDivideCount Register Bit Field Descriptions

Field	Name	Description	Access	Default
31:16	Reserved	–	R/W	–
15:0	–	–	R/W	0x1000

AudioSClkDivideCount Register (AudioSClkDivideCount, offset 0xD8)

The four-byte AudioSClkDivideCount register is writable and contains the value 0x4 at device reset. This register contains a ten-bit counter that specifies the number of cycles of the AUDIO_MCLK for each period of the AUDIO_SCLK.

Table 54: AudioSClkDivideCount Register Bit Field Descriptions

Field	Name	Description	Access	Default
31:10	Reserved	–	R/W	–
9:0	–	–	R/W	0x4

AUDIOLRCKDIVIDECOUNT REGISTER (AUDIOLRCKDIVIDECOUNT, OFFSET 0XDC)

The four-byte AudioLRckDivideCount registers is writable and contains the value 0x100 at device reset. This register contains a ten-bit counter that specifies the number of cycles of the AUDIO_MCLK for each period of the AUDIO_LRCK.

Table 55: AudioLRckDivideCount Register Bit Field Descriptions

Field	Name	Description	Access	Default
31:10	Reserved	–	R/W	–
9:0	–	–	R/W	0x100

AUDIO CONTROL OFFSET REGISTER (AUDIOCONTROL OFFSET 0XE0)

The four-byte Audio Control register is writable and contains a system-dependent value at device reset.

Table 56: Audio Control Offset Register Bit Field Descriptions

Field	Name	Description	Access	Default
31:3	Reserved	–	R/W	–
2	OutEnable (OE)	Out Enable When this field is set to 1, the CODEC_FSYNC pin is configured as an output. When cleared to 0, the CODEC_FSYNC pin is an input. This field is cleared at device reset.	R/W	–
1	Data (DA)	Data When read, this field contains the value of the CODEC_FSYNC pin. When written, the contents of this field are copied to the CODEC_FSYNC pin if the OutEnable field is also set to 1.	R/W	–
0	CntlClk (CC)	Control Clock The contents of the field are written to the AUDIO_CTRL_CLK pin. This field is cleared at device reset.	R/W	–

OFFHOOK WATCHDOG REGISTER (OFFHOOKWATCHDOG, OFFSET 0XE4)

The four-byte OffhookWatchdog register is both readable and writable and contains the value 0x0 at device reset.

Table 57: Offhook Watchdog Register Bit Field Descriptions

Field	Name	Description	Access	Default
31:0	–	When this register is written with a non-zero value, it begins decrementing. When the value reaches 1, the CODEC_OFHK signal is set to 1. If the register contains the value 1 when read, a watchdog reset occurred.	R/W	0

CODEC DESCRIPTOR PROCESSING REGISTERS

The Codec core includes a single DMA processor as described in Section 3: "DMA" on page 31. The following core-specific notes apply to Codec:

- The core does not support the transmit flush operation. The Flush field of the XmtControl register (see "Transmit Channel Control Register (XmtControl, offset 0x200)" on page 32) and the Flushed field of the XmtStatus register (see "Transmit Channel Status Register (XmtStatus, offset 0x20C)" on page 35) are not implemented.
- The core does not provide any receive status header information. There is no minimum value for the RcvOffset field of the RcvControl register (see "Receive Channel Control Register (RcvControl, offset 0x210)" on page 37).
- There is no core-specific storage accessible through the FIFOAddress, FIFODataLow, and FIFODataHigh registers. The Transmit FIFO data, Transmit FIFO pointers, Receive FIFO Data, and Receive FIFO pointers values of the Select field of the FIFOAddress register are not used.

DESCRIPTOR PROCESSOR

The Codec core contains a single descriptor processor that moves data between the core and memory. The descriptor processor is documented in Section 3: "DMA" on page 31. This section describes core-specific attributes of DMA.

DESCRIPTOR BUFFER ALIGNMENT

The Codec core requires all descriptor buffers to be aligned on a four-byte boundary and to be a multiple of four bytes in length.

DESCRIPTOR CORE FLAGS

The descriptor processor provides an eight-bit field in each descriptor for use by individual cores for core-specific purposes. The Codec core does not use any of the Core Flag descriptor bits.

RECEIVE FRAME DATA HEADER SIZE

The Codec core does not return Receive Frame Data Headers. There is no lower bound on the value of the RcvOffset field of the RcvControl register (see "Receive Channel Control Register (RcvControl, offset 0x210)" on page 37).

DMA-RELATED INTERRUPTS AND ERRORS

The Codec core does not have any core-defined DMA-related errors.

INTERRUPTS

RGDT interrupt must be ignored when the core is configured to be in AUDIO or IR mode.

Section 5: Ethernet 10/100 Core

ETHERNET MAC REGISTERS

Table 58: Ethernet MAC Host Interface Register Address Spaces

Address Space	Offset Range
Device and Power Control	0x00-0x1F
Interrupt Control	0x20-0x2F
Reserved	0x30-0x7F
Ethernet MAC Control	0x80-0xAF
Reserved	0xB0-0x1FF
Descriptor Processor	0x200-0x2FF
Reserved	0x300-0x3FF
Ethernet MAC	0x400-0x5FF
Reserved	0x600-0xEFF
System Backplane Configuration Space	0xF00-0xFFF

Table 59: Ethernet MAC Host Interface Registers

Register	Offset	Size	R/W	Reset Value	See...
Device Control – DevControl	0x00	4	R/W	0x0400	62
BIST Status - BISTStatus	0x0C	4	R	0x0	47
Wakeup Length – WakeupLength	0x10	4	R/W	0x0	66
Interrupt Status – IntStatus	0x20	4	R/W	0x0	68
Interrupt Mask – IntMask	0x24	4	R/W	0x0	71
General Purpose Timer – GPTimer	0x28	4	R/W	0x0	73
(BCM4401 B0 only) Magic Packet™ Lower Address - EnetMPAddrLower	0x88	4	R/W	0x0	75
(BCM4401 B0 only) Magic Packet Upper Address - EnetMPAddrUpper	0x8c	4	R/W	0x0	75
Ethernet Filter Table Index – ENetFtAddr	0x90	4	R/W	0x0	76
Ethernet Filter Table Data – ENetFtData	0x94	4	R/W	undefined	77
Ethernet MAC Transmit Maximum Burst Length – EMACXmtMaxBurstLen	0xA0	4	R/W	0x10	79
Ethernet MAC Receive Maximum Burst Length – EMACRcvMaxBurstLen	0xA4	4	R/W	0x10	79
Ethernet MAC Control – EMACControl	0xA8	4	R/W	0x0	79
Ethernet MAC Flow Control – EMACFlowControl	0xAC	4	R/W	0x0	82
Interrupt Receive Lazy Time-out – IntRecvLazy	0x100	4	R/W	0x0	83
Receiver Configuration – RcvConfig	0x400	4	R/W	0x0	85
Receive Max Length – RcvMaxLength	0x404	4	R/W	0x0	86

Table 59: Ethernet MAC Host Interface Registers (Cont.)

Register	Offset	Size	R/W	Reset Value	See...
Transmit Max Length – XmtMaxLength	0x408	4	R/W	0x5EE	86
MII Status/Control – MIIStatusControl	0x410	4	R/W	0x0	86
MII Data – MIIData	0x414	4	R/W	0x0	87
Ethernet Interrupt Mask – EnetIntMask	0x418	4	R/W	0x0	87
Ethernet Interrupt Status – EnetIntStatus	0x41C	4	R/W	0x0	88
CAM Data_L – CAMDataL	0x420	4	R/W	0x0	88
CAM Data_H – CAMDataH	0x424	4	R/W	0x0	89
CAM Control – CAMControl	0x428	4	R/W	0x0	89
Ethernet Control – EnetControl	0x42C	4	R/W	0x0	90
Transmit Control – XmtControl	0x430	4	R/W	0x0	91
Transmit Watermark – XmtWatermark	0x434	4	R/W	0x10	92
MIB Control – MIBControl	0x438	4	R/W	0x0	93
MIB Statistics Counters (see "Management Information Block (MIB) Statistic Counters" on page 94)	0x500-0x5D8	2 or 4	R/W	0x0	94

Table 60: Ethernet PHY MII Registers

Register[a]	Address	Size	R/W	Reset Value	See...
MII Control – MIIControl	0x00	2	R/W	0x3000	98
MII Status – MIIStatus	0x01	2	R/W	0x7809	100
PHY Identifier High – PHYIDHigh	0x02	2	R/W	0x0040	102
PHY Identifier Low – PHYIDLow	0x03	2	R/W	0x61E0	102
Auto-Negotiation Advertisement– AutoNegAdvertise	0x04	2	R/W	0x01E1	103
Auto-Negotiation Link Partner Ability – LinkPartnerAbility	0x05	2	R/W	0x0000	104
Auto-Negotiation Expansion– AutoNegExpansion	0x06	2	R/W	0x0004	105
Auto-Negotiation Next Page – NextPage	0x07	2	R/W	0x2001	106
Auto-Negotiation Link Partner Next Page Transmit – LPNextPage	0x08	2	R/W	0x0000	107
100BASE-X Auxiliary Control – 100BASE-XAuxControl	0x10	2	R/W	0x0000	108
100BASE-X Auxiliary Status – 100BASE-XAuxStatus	0x11	2	R/W	0x0000	110
100BASE-X Receive Error Counter – 100BASE-XRcvErrorCtr	0x12	2	R/W	0x0000	112
100BASE-X Carrier Sense Counter – 100BASE-XCarrierSenseCtr	0x13	2	R/W	0x0000	112
100BASE-X Disconnect Counter – 100BASE-XDisconnectCtr	0x14	2	R/W	0x0200	113
Reserved	0x15	2	R/W	0x0300	N/A
Reserved	0x16	2	R/W	0x0000	N/A
Reserved	0x17	2	R/W	0x0000	N/A
Auxiliary Control/Status – AuxCtlStatus	0x18	2	R/W	0x003x	114
Auxiliary Status Summary – AuxStatusSummary	0x19	2	R/W	0x0000	116
Interrupt – Interrupt	0x1A	2	R/W	0x0F0x	118
Auxiliary Mode 2 – AuxMode2	0x1B	2	R/W	0x008A	119

Table 60: Ethernet PHY MII Registers (Cont.)

Register[a]	Address	Size	R/W	Reset Value	See...
10BASE-T Auxiliary Error and General Status – 10BASE-TErrGenStat	0x1C	2	R/W	0x002x	121
Auxiliary Mode – AuxMode	0x1D	2	R/W	0x000	123
Auxiliary Multiple PHY – AuxMultiPHY	0x1E	2	R/W	0x0000	124
Reserved	0x1F	2	R/W	0x000B	N/A

a. These registers are accessed through the internal MII management interface using the MIIStatusControl and MIIData Ethernet MAC registers.

Table 61: MII Shadow Registers

Register	Address	Size	R/W	Reset Value	See...
Reserved	0x18	2	R/W	0x003x	N/A
Auxiliary Mode 4 – AuxMode4	0x1A	2	R/W	0x0F0x	127
Auxiliary Status 2 – AuxStatus2	0x1B	2	R/W	0x008A	128
Auxiliary Status 3 – AuxStatus3	0x1C	2	R/W	0x002x	129
Auxiliary Mode 3 – AuxMode3	0x1D	2	R/W	0x000	130
Auxiliary Status 4 – AuxStatus4	0x1E	2	R/W	0x0000	130

ETHERNET DEVICE AND POWER CONTROL REGISTERS

The Ethernet device and power control registers include the following:

DEVICE CONTROL REGISTER (DEVCONTROL, OFFSET 0X00)

The four-byte Device Control Register is writable and contains the value 0x0400 at device reset.

This version of the Device Control register applies to the BCM4401 B0 and later controller only

Table 62: Device Control Register Bit Field Descriptions (BCM4401 B0 and Later)

Field	Name	Description	Access	Default
31:18	Reserved	–	R/O	0x0
17	ClkOutputEn (CO)	Clock Output Enable This field enables the rxclk and txclk outputs in MII PHY mode.	R/W	0
16	Reserved	–	R/O	0
15	EPHYReset (ER)	Ethernet PHY Reset When this field is set to 1, the attached internal or external PHY is reset.	R/W	0
14:11	Reserved	–	R/W	0x0
10	InternalEPHY (IP)	Internal Ethernet PHY This read-only field is set to 1 when an on-chip Ethernet PHY is connected to this Ethernet MAC and is cleared to 0 when no on-chip PHY is available.	R/W	1
9:8	Reserved	–	R/W	0x0
7	PatMatchEn (PM)[a]	Pattern Filtering Enable When this field is set to 1, network wakeup frame match filtering on received frames is enabled. When this field is cleared to 0, any received frames that are successfully passed through the Ethernet Frame Processor can cause Power Management Events.	R/W	0
6	MagicPME (MP)	When this field is set to 1, a PME event is generated when a Magic Packet is detected. The event is cleared by writing 0 to this field or clearing the MagicPacket field of the IntStatus register (see "Interrupt Status Register (IntStatus, offset 0x20)" on page 68).	R/W	0
5	LinkPME (LP)	When this field is set to 1, a PME event is generated when a link is detected. The event is cleared by writing 0 to this field or clearing the LinkStatus field of the IntStatus register (see "Interrupt Status Register (IntStatus, offset 0x20)" on page 68).	R/W	0
4:0	Reserved	–	R/W	0x0

a. Magic Packet detection also requires that EnetMPAddrLower (offset 0x88) and EnetMPAddrUpper (offset 0x8c) be programmed for the correct MAC address. See "AMD® Magic Packet Wakeup" on page 147 for additional information.

This version of the Device Control register applies to the rest of the BCM440X family.

Table 63: Device Control Register Bit Field Descriptions (Rest of BCM440X Family)

Field	*Name*	*Description*	*Access*	*Default*
31:23	Reserved	–	R/W	0x0
22:18	PHYAddrReg (PA)	PHY Address Reg This field is used in MII PHY mode as the address bits receiving or transmitting frames on the serial management MDC/MDIO interface.	R/W	0
17	ClkOutputEn (CO)	Clock Output Enable This field enables the rxclk and txclk outputs in MII PHY mode.	R/W	0
16	MIIPHYModeEn (MP)	MII PHY Mode Enable When this field is set to 1, the ethernet core acts as a PHY-layer device that can be attached to the MII interface of an external MAC. When this field is cleared to 0, the core acts as an Ethernet MAC that can control a PHY device using the MII interface.	R/W	0
15	EPHYReset (ER)	Ethernet PHY Reset When this field is set to 1, the attached internal or external PHY is reset.	R/W	0x0
14:11	Reserved	–	R/W	1
10	InternalEPHY (IP)	Internal Ethernet PHY This read-only field is set to 1 when an on-chip Ethernet PHY is connected to this Ethernet MAC and is cleared to 0 when no on-chip PHY is available.	R/W	0x0
9:8	Reserved	–	R/W	0
7	PatMatchEn (PM)	Pattern Filtering Enable When this field is set to 1, network wakeup frame match filtering on received frames is enabled. When this field is cleared to 0, any received frames that are successfully passed through the Ethernet Frame Processor can cause Power Management Events.	R/W	0
6:0	Reserved	–	R/W	0x0

BUILT-IN SELF TEST REGISTER (BISTSTATUS, OFFSET 0X0C)

The four-byte, read-only BISTStatus register contains the value 0x0 at device reset.

This version of the Built-In Self Test register applies to the BCM4401 B0 and later controller only.

Table 64: BIST Status Register Bit Field Descriptions (BCM4401 B0 and Later)

Field	Name	Description	Access	Default
31:12	Reserved	–	RO	0
11:0	BISTStatus (BS)	Built-In Self Test Status This field contains the result of the Built-In Self Test for all of the RAMs in the Ethernet core. If any bit is set to 1, the BIST failed for the corresponding RAM. See Table 46 or the mapping of RAMs to bits within this field. This field is valid only when the BISTDone field is set in the SBTMSTATEHIGH register (see "System Backplane Target State High Register (SBTMSTATEHIGH, offset 0xF9C)" on page 18).	RO	0

Table 65: RAM-to-Bit Mapping (BCM4401 B0 and Later)

Bit	RAM
11	Transmit Dual Port
10	Transmit DMA FIFO
9	Receive Dual Port
8	Receive DMA Pointer
7	Receive DMA FIFO
6	Wakeup Patterns
5	MIB Data
4	EMAC Transmit Data
3	CAM
2	Receive SL FIFO B
1	Receive SL FIFO A
0	EMAC Receive Data

This version of the Built-In Self Test register applies to the rest of the BCM440X family.

Table 66: BIST Status Register Bit Field Descriptions (Rest of BCM440X Family)

Field	Name	Description	Access	Default
31:2	Reserved	–	RO	0
1:0	BISTStatus (BS)	Built-In Self Test Status This field contains the result of the Built-In Self Test for all of the RAMs in the Ethernet core. If any bit is set to 1, the BIST failed for the corresponding RAM. See Table 46 or the mapping of RAMs to bits within this field. This field is valid only when the BISTDone field is set in the SBTMSTATEHIGH register (see "System Backplane Target State High Register (SBTMSTATEHIGH, offset 0xF9C)" on page 18).	RO	0

Table 67: RAM-to-Bit Mapping (Rest of BCM440X Family)

Bit	RAM
0	Receive DMA FIFO
1	Transmit DMA FIFO

WAKEUP LENGTH REGISTER (WAKEUPLENGTH, OFFSET 0X10)

The four-byte Wakeup Length register specifies the length of the power management wakeup patterns.

Table 68: Wakeup Length Register Bit Field Descriptions

Field	Name	Description	Access	Default
31	Disable3 (D3)	Disable Pattern 3 When this field is set to 1, the device Ethernet does not try to match wakeup pattern 3 with incoming frames. When cleared to 0, incoming frames are matched against wakeup pattern 3.	R/W	–
30:24	Pattern3 (P3)	Pattern 3 This field specifies one less than the number of valid bytes in wakeup pattern 3. In order to match wakeup pattern 3, the length of an incoming frame must be greater than the value of this field.	R/W	–
23	Disable2 (D2)	Disable Pattern 2 When this field is set to 1, the device Ethernet does not try to match wakeup pattern 2 with incoming frames. When cleared to 0, incoming frames are matched against wakeup pattern 2.	R/W	–
22:16	Pattern2 (P2)	Pattern 2 This field specifies one less than the number of valid bytes in wakeup pattern 2. In order to match wakeup pattern 2, the length of an incoming frame must be greater than the value of this field.	R/W	–
15	Disable1 (D1)	Disable Pattern 1 When this field is set to 1, the device Ethernet does not try to match wakeup pattern 1 with incoming frames. When cleared to 0, incoming frames are matched against wakeup pattern 1.	R/W	–
14:8	Pattern1 (P1)	Pattern 1 This field specifies one less than the number of valid bytes in wakeup pattern 1. In order to match wakeup pattern 1, the length of an incoming frame must be greater than the value of this field.	R/W	–
7	Disable0 (D0)	Disable Pattern 0 When this field is set to 1, the device Ethernet does not try to match wakeup pattern 0 with incoming frames. When cleared to 0, incoming frames are matched against wakeup pattern 0.	R/W	–
6:0	Pattern0 (P0)	Pattern 0 This field specifies one less than the number of valid bytes in wakeup pattern 0. In order to match wakeup pattern 0, the length of an incoming frame must be greater than the value of this field. See "Network Wakeup Frame Detection" on page 146 for details.	R/W	–

ETHERNET INTERRUPT HANDLING REGISTERS

The Ethernet interrupt handling registers include the following:

INTERRUPT STATUS REGISTER (INTSTATUS, OFFSET 0X20)

The four-byte Interrupt Status register is both readable and writable and contains the status of all device Ethernet interrupt sources. A bit set in this register indicates that an interrupt has been generated. The interrupt signal is not asserted unless the corresponding enable bit in the IntMask register is set to 1 (see "Interrupt Mask Register (IntMask, offset 0x24)" on page 71). This register has the value 0x0 at device reset. See "Error Conditions" on page 47 or more information on error conditions reported by the descriptor processor.

This version of the Interrupt Status register applies to the BCM4401 B0 and later controller only.

Table 69: Interrupt Status Register Bit Field Descriptions (BCM4401 B0 and Later)

Field	Name	Description	Access	Default
31:27	Reserved	–	R/W	0
26	EMACInterrupt (EI)	Ethernet MAC Interrupt When this field is set to 1, it indicates that an Ethernet interrupt (MII, MIB, or Flow Control) has occurred.	R/W	0
25	Reserved	–	R/W	0
24	XmtInt (XI)	Transmit Notification Interrupt This field contains a transmit notification interrupt bit for the channel. This field is set to 1 when a transmit completion interrupt is generated for the channel. Writing a 1 to this field clears both the field and the pending interrupt. Disabling the transmit channel does not clear this interrupt.	R/W	0
23:17	Reserved	–	R/W	0
16	RcvInt (RI)	Receive Notification Interrupt This field contains a receive notification interrupt bit for the channel. This field is set to 1 when a receive completion interrupt or a receive lazy interrupt is generated for the channel. Writing a 1 this field clears both the field and the pending interrupt. Disabling the receive channel does not clear this interrupt.	R/W	0
15	XmtFIFOUf (XU)	Transmit FIFO Underflow This field is set to 1 when a channel cannot process an outgoing frame because no more data are available in the transmit FIFO. This field is cleared to 0 by resetting the transmit channel.	R/W	0
14	RcvFIFOOf (RO)	Receive FIFO Overflow This field is set to 1 when a frame is being received and no space is available in the receive FIFO. This interrupt is cleared to 0 by resetting the receive channel.	R/W	0
13	RcvDescUf (RU)	Receive Descriptor Underflow This field is set to 1 when the descriptor processor reports a receive descriptor underflow condition. This interrupt is cleared to 0 by supplying descriptors to the receive channel. If no descriptors are provided before the receive FIFO becomes full, a receive FIFO overflow error and interrupt occurs.	R/W	0
12	DescProtoErr (DP)	Descriptor Protocol Error This field is set to 1 when a descriptor processor reports a descriptor protocol error. This interrupt is cleared to 0 by resetting the channel that caused the error.	R/W	0

Table 69: Interrupt Status Register Bit Field Descriptions (BCM4401 B0 and Later) (Cont.)

Field	Name	Description	Access	Default
11	DataErr (DA)	Data Error This field is set to 1 when the descriptor processor reports a data transfer error. This interrupt is cleared to 0 by resetting the channel that caused the error.	R/W	0
10	DescrErr (DE)	Descriptor Error This field is set to 1 when the descriptor processor reports a descriptor read error. This interrupt is cleared to 0 by resetting the channel that caused the error.	R/W	0
9:8	Reserved	–	R/W	0
7	TimeOut (TO)	Timeout This field is set to 1 when a general purpose timer timeout interrupt is generated. Writing a 1 to this field clears both the field and the pending interrupt.	R/W	0
6	PME (PM)	Power Management Event This field is set to 1 when a power management event is generated. Writing a 1 to this field clears both the field and the pending interrupt.	R/W	0
5	LinkStatus (LS)	his field is set to 1 when a link status change interrupt occurs. Writing a 1 to this field clears both the field and the pending interrupt.	R/W	0
4:0	Reserved	–	R/W	0

This version of the Interrupt Status register applies to the rest of the BCM440X family.

Table 70: Interrupt Status Register Bit Field Descriptions (Rest of BCM440X Family)

Field	Name	Description	Access	Default
31:29	Reserved	–	R/W	0
28	IntMIIRead (IR)	Interrupt MII Read This field is set to 1 when an MDC/MDIO register read occurs. Writing a 1 to this field clears both the field and the pending interrupt.	R/W	0
27	IntMIIWrite (IW)	Interrupt MII Write This field is set to 1 when an MDC/MDIO register write occurs. Writing a 1 to this field clears both the field and the pending interrupt.	R/W	0
26	EMACInterrupt (EI)	Ethernet MAC Interrupt When this field is set to 1, it indicates that an Ethernet interrupt (MII, MIB, or Flow Control) has occurred.	R/W	0
25	Reserved		R/W	0
24	XmtInt (XI)	Transmit Notification Interrupt This field contains a transmit notification interrupt bit for the channel. This field is set to 1 when a transmit completion interrupt is generated for the channel. Writing a 1 to this field clears both the field and the pending interrupt. Disabling the transmit channel does not clear this interrupt.	R/W	0
23:17	Reserved	–	R/W	0

Table 70: Interrupt Status Register Bit Field Descriptions (Rest of BCM440X Family) (Cont.)

Field	Name	Description	Access	Default
16	RcvInt (RI)	Receive Notification Interrupt This field contains a receive notification interrupt bit for the channel. This field is set to 1 when a receive completion interrupt or a receive lazy interrupt is generated for the channel. Writing a 1 this field clears both the field and the pending interrupt. Disabling the receive channel does not clear this interrupt.	R/W	0
15	XmtFIFOUf (XU)	Transmit FIFO Underflow This field is set to 1 when a channel cannot process an outgoing frame because no more data are available in the transmit FIFO. This field is cleared to 0 by resetting the transmit channel.	R/W	0
14	RcvFIFOOf (RO)	Receive FIFO Overflow This field is set to 1 when a frame is being received and no space is available in the receive FIFO. This interrupt is cleared to 0 by resetting the receive channel.	R/W	0
13	RcvDescUf (RU)	Receive Descriptor Underflow This field is set to 1 when the descriptor processor reports a receive descriptor underflow condition. This interrupt is cleared to 0 by supplying descriptors to the receive channel. If no descriptors are provided before the receive FIFO becomes full, a receive FIFO overflow error and interrupt occurs.	R/W	0
12	DescProtoErr (DP)	Descriptor Protocol Error This field is set to 1 when a descriptor processor reports a descriptor protocol error. This interrupt is cleared to 0 by resetting the channel that caused the error.	R/W	0
11	DataErr (DA)	Data Error This field is set to 1 when the descriptor processor reports a data transfer error. This interrupt is cleared to 0 by resetting the channel that caused the error.	R/W	0
10	DescrErr (DE)	Descriptor Error This field is set to 1 when the descriptor processor reports a descriptor read error. This interrupt is cleared to 0 by resetting the channel that caused the error.	R/W	0
9:8	Reserved	–	R/W	0
7	TimeOut (TO)	Timeout This field is set to 1 when a general purpose timer timeout interrupt is generated. Writing a 1 to this field clears both the field and the pending interrupt.	R/W	0
6	PME (PM)	Power Management Event This field is set to 1 when a power management event is generated. Writing a 1 to this field clears both the field and the pending interrupt.	R/W	0
5:0	Reserved	–	R/W	0

INTERRUPT MASK REGISTER (INTMASK, OFFSET 0X24)

The four-byte Interrupt Mask register is both readable and writable and selectively enables the delivery of interrupts to the host processor. The enables in this register do not affect the recording of interrupt conditions in the IntStatus register; they affect only the propagation of the interrupt out of the core. This register has a value of 0x0 at device reset.

Table 71: Interrupt Mask Register Bit Field Descriptions

Field	*Name*	*Description*	*Access*	*Default*
31:29	Reserved	–	R/W	0
28	IntMIIRead (IR)	Interrupt MII Read When set to 1, this field enables MII read interrupts.	R/W	0
27	IntMIIWrite (IW)	Interrupt MII Write When set to 1, this field enables MII write interrupts.	R/W	0
28:27	Reserved (BCM4401 B0 only)		R/W	0
26	EMACInterrupt (EI)	Ethernet MAC Interrupt When this field is set to 1, it enables interrupt delivery for the Ethernet MAC. The interrupt is turned off by clearing this field to 0.	R/W	0
25	Reserved	–	R/W	0
24	XmtIntEn (XI)	Transmit Interrupt Enable When set to 1, this field enables interrupt delivery from the transmit channel of the descriptor processor.	R/W	0
23:17	Reserved	–	R/W	0
16	RcvIntEn (RI)	Receive Interrupt Enable When set to 1, this field enables interrupt delivery from the receive channel of the descriptor processor.	R/W	0
15	XmtUfEn (XU)	Transmit Underflow Enable When set to 1, this field enables interrupts that occur when a channel cannot process an outgoing frame because no more data are available in the transmit FIFO.	R/W	0
14	RcvOfEn (RO)	Receive Overflow Enable When set to 1, this field enables interrupts that occur when a channel cannot process an incoming frame because no space is available in the receive data FIFO.	R/W	0
13	RcvUfEn (RU)	Receive Underflow Enable When set to 1, this field enables interrupts that occur when a channel cannot process an incoming frame because no descriptors are available.	R/W	0
12	DescProtoErrEn (DP)	Descriptor Protocol Error Enable When set to 1, this field enables interrupts on descriptor programming errors.	R/W	0
11	DataErrEn (DA)	Data Error Enable When set to 1, this field enables interrupts on errors that occur while transferring data to or from memory.	R/W	0
10	DescErrEn (DE)	Descriptor Error Enable When set to 1, this field enables interrupts on descriptor read errors.	R/W	0
9:8	Reserved	–	R/W	0

Table 71: Interrupt Mask Register Bit Field Descriptions (Cont.)

Field	Name	Description	Access	Default
7	TimeOut (TO)	Timeout When set to 1, this field enables general purpose timer timeout interrupts.	R/W	0
6	PME (PM)	Power Management Event When set to 1, this field enables PME interrupts.	R/W	0
5	Link Status (LS) (BCM4401 B0 only)	When set to 1, this field enables interrupts on link status changes.	R/W	0
5:0	Reserved	–	R/W	0

GENERAL PURPOSE TIMER REGISTER (GPTIMER, OFFSET 0X28)

The four-byte General Purpose Timer register is a free-running 32-bit counter that decrements on the 62.5 MHz clock, giving a timeout range of about 16 ns to 70 ns. This register is both readable and writable and has the value 0x0 at device reset.

Table 72: General Purpose Timer Register Bit Field Descriptions

Field	*Name*	*Description*	*Access*	*Default*
31:0	–	Writing to this register initializes the time-out counter to the value written. When the counter decrements to 0, a general purpose timer time-out interrupt is generated. The counter is then loaded with the last value written and continues to decrement. Reading the register returns the current value of the timer. Writing a 0 to this register disables the counter.	R/W	0

ETHERNET ADDRESS AND DATA REGISTERS

The Ethernet address and data registers include the following:

ETHERNET MAGIC PACKET™ ADDRESS LOWER REGISTER (ENETMPADDRLOWER, OFFSET 0X88) (BCM4401 B0 ONLY)

The four-byte Ethernet Magic Packet Address Lower register is both readable and writable and contains the value 0x0 at device reset. This register must be initialized with the lower four bytes of the Ethernet MAC address of this core before magic-packet recognition can be enabled.

Table 73: Ethernet Magic Packet Address Lower Register Descriptions (BCM4401 B0 and Later)

Field	Name	Description	Access	Default
31:0	MAC Address Lower	This field contains the lower four bytes of the MAC address of this core.	R/W	0

ETHERNET MAGIC PACKET ADDRESS UPPER REGISTER (ENETMPADDRUPPER, OFFSET 0X8C) (BCM4401 B0 ONLY)

The four-byte Ethernet Magic Packet Address Upper register is both readable and writable and contains the value 0x0 at device reset. This register must be initialized with the upper two bytes of the Ethernet MAC address of this core before Magic Packet recognition can be enabled.

Table 74: Ethernet Magic Packet Address Upper Register Descriptions (BCM4401 B0 and Later)

Field	Name	Description	Access	Default
31:16	Reserved	–	RO	0
15:0	MAC Address Upper	This field contains the upper two bytes of the MAC address of this core.	R/W	0

Programming example for the ENetMPAddrLower and ENetMPAddrUpper registers:

```
/* MAC Address: 00-01-02-03-04-05 */

mac[6] = 0x000102030405

RegWrite(ENetMPAddrLower, ((mac[2]<<24) | (mac[3]<<16) | (mac[4]<<8) | mac[5]);
RegWrite(ENetMPAddrUpper, ((mac[0]<<8) | mac[1]);
```

ETHERNET FILTER TABLE ADDRESS REGISTER (ENETFTADDR, OFFSET 0X90)

The four-byte Ethernet Filter Table Address register is both readable and writable and contains the value 0x0 at device reset.

Table 75: Ethernet Filter Table Address Register Bit Field Descriptions

Field	Name	Description	Access	Default
31:10	Reserved	–	R/W	0
9:0	–	This field contains a 10-bit address offset into the Ethernet filter table. Subsequent accesses to the ENetFtData register (see "Ethernet Filter Table Data Register (ENetFtData, offset 0x94)" on page 77) read or write the four bytes of the filter table at this address. Ethernet wakeup filter patterns and match enable bits are used to generate wakeup events when the device is in a powered-down state. See "Wakeup Frame Patterns" on page 146 for more information on wakeup patterns. The filter table address space is shown in Table 76.	R/W	0

Table 76: Ethernet Filter Table Address Space

Item	Address
Wakeup Pattern	0p pxxx xx--
Pattern Enable	1- --pp xx--

In Table 76, an *x* represents a bit used to address an individual entry within a larger group, an *h* distinguishes the high and low half of an eight-byte entry, and a *p* represents a pattern number. The low two bits of the address are ignored and always read back as 0.

ETHERNET FILTER TABLE DATA REGISTER (ENETFTDATA, OFFSET 0X94)

The four-byte Ethernet Filter Table Data register is a port into the filter table entry addressed by the contents of the ENetFtAddr register (see "Ethernet Filter Table Address Register (ENetFtAddr, offset 0x90)" on page 76).

Table 77: Ethernet Filter Table Data Register Bit Field Descriptions

Field	Name	Description	Access	Default
31:0	–	Reading or writing this register reads or writes the four bytes of the filter table entry referenced by the contents of the ENetFtAddr register (see "Ethernet Filter Table Address Register (ENetFtAddr, offset 0x90)" on page 76). See "Wakeup Frame Patterns" on page 146 for more information on wakeup patterns. Updating the filter table generally requires multiple writes of the ENetFtData register with intervening writes to the ENetFtAddr register (see "Ethernet Filter Table Data Register (ENetFtData, offset 0x94)" on page 77). During the sequence of writes, the table can be in a state that is not meaningful. Driver software can disable address filtering while updating the filter table. The contents of all entries in the Ethernet Filter Table are undefined at device reset.	R/W	not defined

ETHERNET MEDIUM ACCESS CONTROL REGISTERS

The Ethernet medium access control registers include the following:

ETHERNET MAC TRANSMIT MAX BURST LENGTH REGISTER (EMACXMTMAXBURSTLEN, OFFSET 0XA0)

The four-byte Ethernet MAC Max Transmit Burst Length register is both readable and writable and contains the value 0x10 on device reset.

ETHERNET MAC RECEIVE MAX BURST LENGTH REGISTER (EMACRCVMAXBURSTLEN, OFFSET 0XA4)

The four-byte Ethernet MAC Max Receive Burst Length register is both readable and writable and contains the value 0x10 on device reset.

ETHERNET MAC CONTROL REGISTER (EMACCONTROL, OFFSET 0XA8)

The four-byte Ethernet MAC Control register is both readable and writable and contains the value 0x0 at device reset.

Table 78: Ethernet MAC Control Register Bit Field Descriptions

Field	*Name*	*Description*	*Access*	*Default*
31:8	Reserved	–	R/W	0
7:5	LEDControl (LC)	LED Control The values in the three bits in this field select how the LEDs are assigned for the Ethernet function. See the LED Control Select tables below for details.	R/W	0
4	Reserved	–	R/W	0
3	EPHYEnergyDetect (ED)	EPHY Energy Detect This field is set to 1 when the attached on-chip PHY detects energy on the physical medium. The hardware never clears this field; it is cleared by writing a 0 to it. This field is reserved when no on-chip PHY is attached to this MAC.	R/W	0
2	EPHYPowerDown (EP)	EPHY Power Down When this field is set to 1, the attached on-chip PHY is powered off for maximum power savings. This field is reserved when no on-chip PHY is attached to this MAC.	R/W	0
1	Reserved	–	R/W	0
0	CRCCheckXmt (CC)	CRC Check Transmit When this field is set to 1, the Ethernet MAC does a CRC check on transmit packets.	R/W	0

LED CONTROL SELECT TABLES

Table 79: LED Control 000 Selects

196 FBGA	128 PQFP	LED	Output Condition
G12	78	LED3	LED 3 Ethernet collision
G14	77	LED2	LED 2 Ethernet speed
H13	76	LED1	LED 1 Ethernet activity (transmit and receive)
G13	75	LED0	LED 0 Ethernet link integrity

Table 80: LED Control 001 Selects

196 FBGA	128 PQFP	LED	Output Condition
G12	78	LED3	LED 3 Ethernet collision
G14	77	LED2	LED 2 Ethernet activity (high speed)
H13	76	LED1	LED 1 Ethernet activity (low speed)
G13	75	LED0	LED 0 Ethernet link integrity

Table 81: LED Control 010 Selects

196 FBGA	128 PQFP	LED	Output Condition
G12	78	LED3	LED 3 Ethernet collision
G14	77	LED2	LED 2 Ethernet receive
H13	76	LED1	LED 1 Ethernet transmit
G13	75	LED0	LED 0 Ethernet link integrity

Table 82: LED Control 011 Selects

196 FBGA	128 PQFP	LED	Output Condition
G12	78	LED3	LED 3 Ethernet speed
G14	77	LED2	LED 2 Ethernet receive
H13	76	LED1	LED 1 Ethernet transmit
G13	75	LED0	LED 0 Ethernet link integrity

Table 83: LED Control 100 Selects

196 FBGA	128 PQFP	LED	Output Condition
G12	78	LED3	LED 3 Ethernet collision
G14	77	LED2	LED 2 Ethernet transmit
H13	76	LED1	LED 1 Ethernet receive
G13	75	LED0	LED 0 Ethernet speed

Table 84: LED Control 101 Selects

196 FBGA	*128 PQFP*	*LED*	*Output Condition*
G12	78	LED3	LED 3 Ethernet collision
G14	77	LED2	LED 2 Ethernet speed
H13	76	LED1	LED 1 Ethernet receive
G13	75	LED0	LED 0 Ethernet link integrity

Table 85: LED Control 110 Selects

196 FBGA	*128 PQFP*	*LED*	*Output Condition*
G12	78	LED3	LED 3 Ethernet collision
G14	77	LED2	LED 2 Ethernet speed
H13	76	LED1	LED 1 Ethernet transmit
G13	75	LED0	LED 0 Ethernet link integrity

Table 86: LED Control 111 Selects

196 FBGA	*128 PQFP*	*LED*	*Output Condition*
G12	78	LED3	LED 3 Ethernet collision
G14	77	LED2	LED 2 Ethernet activity (transmit and receive)
H13	76	LED1	LED 1 Ethernet link integrity (100 Mbps)
G13	75	LED0	LED 0 Ethernet link integrity (10 Mbps)

ETHERNET MAC FLOW CONTROL REGISTER (EMACFLOWCONTROL, OFFSET 0XAC)

The four-byte Ethernet MAC Flow Control register is both readable and writable and contains the value 0x0 at device reset.

Table 87: Transmit FIFO Control Register Bit Field Descriptions

Field	*Name*	*Description*	*Access*	*Default*
31:16	Reserved	–	R/W	0
15	PauseEn (PE)	Pause Enable When set to 1, this field enables a pause in frame generation.	R/W	0
14:8	Reserved	–	R/W	0
7:0	RcvFlowControl (RF)	Receiver Flow Control The value written in this field forces the receiver to flow control after the data in the receive FIFO reaches this level.	R/W	0

RECEIVE INTERRUPT CONTROL REGISTERS

This section describes the receive interrupt registers.

INTERRUPT RECEIVE LAZY TIME-OUT REGISTER (INTRECVLAZY, OFFSET 0X100)

The four-byte Interrupt Receive Lazy Time-out register is both readable and writable and has the value 0x0 at device reset.

Table 88: Interrupt Receive Lazy Time-out Register Bit Field Descriptions

Field	*Name*	*Description*	*Access*	*Default*
31:24	FrameCount (FC)	Frame Count This field contains the number of frames that must be received before a lazy interrupt is generated. A value of 0x0 disables the frame count interrupt.	R/W	0
23:0	TimeOut (TO)	Timeout This field contains the receive interrupt time-out value in backplane clock cycles. An interrupt is generated when TimeOut (TO) backplane clock cycles have passed since the first frame. A value of 0x0 disables the time-out interrupt.	R/W	0

Updates to each field in this register take effect immediately if the field's value changes from zero to a non-zero value. Otherwise, the update does not take effect until the next time a frame is received after a receive interrupt is generated on this channel.

ETHERNET DESCRIPTOR PROCESSING REGISTERS

The Ethernet MAC core includes a single DMA processor as described in Section 3: "DMA" on page 31. The following core-specific notes apply to the Ethernet MAC:

- The core does not support the transmit flush operation. The Flush field of the XmtControl register (see "Transmit Channel Control Register (XmtControl, offset 0x200)" on page 32) is not implemented.
- The value of the RcvOffset field of the RcvControl register (see "Receive Channel Control Register (RcvControl, offset 0x210)" on page 37) must be greater than or equal to 28.

ETHERNET MAC REGISTERS

The Ethernet MAC registers include the following:

RECEIVER CONFIGURATION REGISTER (RCVCONFIG, OFFSET 0X400)

The four-byte Receive Configuration register is both readable and writable and is initialized to 0x0 at device reset.

Table 89: Receiver Configuration Register Bit Field Descriptions

Field	*Name*	*Description*	*Access*	*Default*
31:8	Reserved	–	R/W	0
7	RejectFilter (RF)	Reject Filter When this field is set to 1, incoming unicast frames with destination MAC addresses that match in the address CAM are discarded and those that do not match are accepted. Incoming multicast (non-broadcast) frames are discarded if they match in the CAM or if the AccMulti field of the RcvConfig register field is set. When this field is cleared to 0, unicast frames that match in the CAM are accepted and frames that do not match are discarded. Multicast frames are accepted if they match in the CAM or if the AccMulti field of the RcvConfig register is set. This field does not affect the reception of broadcast frames.	R/W	0
6	UniFlow (UF)	Unicast Flow Control Frame When this field is set to 1, the MAC accepts Unicast addressed PAUSE frames which match a CAM entry. When this field is cleared to 0, the MAC only accepts PAUSE frames that contain a DA that matches the unique multicast address defined in 802.3x. This field is only valid when the MAC flow control is enabled (EnFlow = 1).	R/W	0
5	EnFlow (EF)	Enable Flow Control When this field is set to 1, if a MAC Pause frame is received, the receiver indicates to the transmitter to cease its transmissions for the time specified in the Pause frame. This field is only valid when the MAC is in full-duplex mode.	R/W	0
4	Lpbk (LB)	Loopback When this field is set to 1, all data transmitted by the MAC internally is received by the MAC receiver. The MII signals COL,CRS, RX_DV, RXD, and RX_ER are ignored by the MAC.	R/W	0
3	Prom (PR)	Promiscuous When this field is set to 1, all frames are accepted by the receiver.	R/W	0
2	RcvDisableTx (RD)	Receive Disable while Transmitting When this field is set to 1, the receiver is inhibited from receiving any frames while the MAC transmitter is transmitting. This field is only valid if the MAC is in full-duplex mode.	R/W	0
1	AccMulti (AM)	Accept All Multicast When this field is set to 1, all multicast frames are accepted.	R/W	0
0	DisB (DB)	Disable Broadcast When this field is set to 1, all broadcast frames are rejected. If the receiver is in promiscuous mode, the value of this field is ignored and all fields are accepted.	R/W	0

RECEIVE MAXIMUM LENGTH REGISTER (RCVLENGTH, OFFSET 0X404)

The four-byte Receive Maximum Length register is both readable and writable and is initialized to 0x0 at device reset.

Table 90: Receive Maximum Length Register Bit Field Descriptions

Field	Name	Description	Access	Default
31:11	Reserved	–	R/W	0
10:0	RcvMaxLen (RM)	Receive Maximum Length The value stored in this register specifies the largest valid Ethernet frame to be received. Frames that have lengths larger than this value cause the Large Frame field in the receive status word to be set. In addition, the oversize or jabber MIB counter is incremented.	R/W	0

TRANSMIT MAXIMUM LENGTH REGISTER (XMTMAXLENGTH, OFFSET 0X408)

The four-byte Transmit Maximum Length register is both readable and writable and is initialized to 0x5EE at device reset.

Table 91: Transmit Maximum Length Register Bit Field Descriptions

Field	Name	Description	Access	Default
31:11	Reserved	–	R/W	–
10:0	XmtMaxLen (TM)	Transmit Maximum Length The value written to this register sets the maximum length for a transmit packet. This counter is used in the statistics counter block. This register is set to 1518 after reset. This register should only be updated while the transmitter is disabled.	R/W	0x5EE

MII STATUS/CONTROL REGISTER (MIISTATUSCONTROL, OFFSET 0X410)

The four-byte MII Status/Control register is both readable and writable and is initialized to 0x0 at device reset.

Table 92: MII Status/Control Register Bit Field Descriptions

Field	Name	Description	Access	Default
31:8	Reserved	–	R/W	0
7	PreEn (PR)	Preamble Enable When this field is set to 1, a preamble sequence is prepended to each MII management read or write operation. The preamble consists of 32 consecutive 1s.	R/W	0
6:0	MDC (MD)	MDC Frequency A value of 0 turns off the MDC clock. Writing a value greater than 0 to this field causes the MDC clock to turn on. The frequency of the clock is 1 / (100MHz * MDC[6:0]).	R/W	0

MII DATA REGISTER (MIIDATA OFFSET 0X414)

The four-byte MII Data register is both readable and writable and is initialized to 0x0 at device reset.

Table 93: MII Data Register Bit Field Descriptions

Field	Name	Description	Access	Default
31:30	StartBits (SB)	Start Bits The value 01 indicates a start of frame to the MII slave devices.	R/W	0
29:28	Opcode (OP)	Opcode This field contains the opcode. An opcode of <01> is a write; an opcode of 10 is a read.	R/W	0
27:23	PhysMedia (PM)	Physical Media Device This field is the 5-bit address of the MII slave device.	R/W	0
22:18	RegAddr (RA)	Register Address This field is the five-bit address of the register within the specified PMD device.	R/W	0
17:16	Turnaround (TA)	Turnaround This field contains the turnaround value. A valid turn around value is 10.	R/W	0
15:0	Data (D)	Data On a read operation, this field contains the value read from the MII slave device. Following the completion of a write operation, this field contains the original value written to the MII slave device.	R/W	0

ETHERNET INTERRUPT MASK REGISTER (ENETINTMASK, OFFSET 0X418)

The four-byte Ethernet Interrupt Mask register is both readable and writable and is initialized to 0x0 at device reset.

Table 94: Ethernet Interrupt Mask Register Bit Field Descriptions

Field	Name	Description	Access	Default
31:3	Reserved	–	R/W	0
2	FlowMask (FM)	Flow Mask Writing a 1 to this field enables the flow control interrupt.	R/W	0
1	MIBMask (MB)	MIB Mask Writing a 1 to this field enables the MIB interrupt.	R/W	0
0	MIIMask (MI)	MII Mask Writing a 1 to this field enables the MII interrupt.	R/W	0

ETHERNET INTERRUPT STATUS REGISTER (ENETINTSTATUS, OFFSET 0X41C)

The four-byte Ethernet Interrupt Status register is both readable and writable and is initialized to 0x0 at device reset.

Table 95: Ethernet Interrupt Status Register Bit Field Descriptions

Field	Name	Description	Access	Default
31:3	Reserved	–	R/W	0
2	FlowInt (FM)	Flow Interrupt When the value in this field is 1 and the corresponding mask field in the EnetIntMask register (see "Ethernet Interrupt Mask Register (EnetIntMask, offset 0x418)" on page 87) is set to 1, an interrupt to the processor occurs. The interrupt line remains asserted until the FlowInt field is cleared. To clear the interrupt, write a 1 to this field.	R/W	0
1	MIBInt (MB)	MIB Interrupt When the value in this field is 1 and the corresponding mask field in the EnetIntMask register (see "Ethernet Interrupt Mask Register (EnetIntMask, offset 0x418)" on page 87) is set to 1, an interrupt to the processor occurs. The interrupt line remains asserted until the MIBInt field is cleared. To clear the interrupt, write a 1 to this field.	R/W	0
0	MIIInt (MI)	MII Interrupt When the value in this field is 1, and the corresponding mask field in the EnetIntMask register (see "Ethernet Interrupt Mask Register (EnetIntMask, offset 0x418)" on page 87) is set to 1, an interrupt to the processor occurs. The interrupt line remains asserted until the MIIInt field is cleared. To clear the interrupt, write a 1 to this field.	R/W	0

CAM DATA LOW REGISTER (CAMDATAL, OFFSET 0X420)

The four-byte CAM Data Low register is both readable and writable and is initialized to 0x0 at device reset.

Table 96: CAM Data Low Register Bit Field Descriptions

Field	Name	Description	Access	Default
31:0	CAMDataL (CD)	CAM Data Low When performing a CAM write or CAM mask write operation, the data stored in this register is used as the lower 32 bits of a 48-bit data field. After completion of a read operation, this register contains the lower 32 bits of data read from the CAM.	R/W	0

CAM DATA HIGH REGISTER (CAMDATAH, OFFSET 0X424)

The four-byte CAM Data High register is both readable and writable and is initialized to 0x0 at device reset.

Table 97: CAM Data High Register Bit Field Descriptions

Field	Name	Description	Access	Default
31:17	Reserved	–	R/W	0
16	ValidBit (VB)	Valid Bit When this field is cleared to 0, the address is invalid and is not used for address comparisons by the CAM. Setting this field to 1 causes the address to be compared to the destination address of the next incoming frame.	R/W	0
15:0	CAMDataH (CD)	CAM Data High When performing a CAM write or CAM mask write operation, the data stored in this register is used as the upper 16 bits of a 48-bit data field. Following the completion of a read operation, this register contains the upper 16 bits of data read from the CAM.	R/W	0

CAM CONTROL REGISTER (CAMCONTROL, OFFSET 0X428)

The four-byte CAM Control register is both readable and writable and is initialized to 0x0 at device reset.

Table 98: CAM Control Register Bit Field Descriptions

Field	Name	Description	Access	Default
31	CAMBusy (CB)	CAM Busy Writing to the CAMControl register initiates a CAM read/write cycle and sets this field to 1. After completion of the CAM read/write cycle, this field is cleared to 0. This field is read only.	R/W	0
30:22	Reserved	–	R/W	0
21:16	Index[5:0] (IX)	Index [6] This field contains one of the 64 possible indexes into the CAM. The value of this field is ignored if the MaskSelect field is set to 1.	R/W	0
15:4	Reserved	–	R/W	0
3	CAMWrite (CW)	CAM Write Set this field to 1 to perform a CAM write cycle.	R/W	0
2	CAMRead (CR)	CAM Read Set this field to 1 to perform a CAM read cycle.	R/W	0
1	MaskSelect (MS)	Mask Select When this field is cleared to 0, the value in the Index[5:0] field is used as the entry index into the CAM. When this field is set to 1, the CAM read or write operation is directed to the 48-bit mask register.	R/W	0
0	CAMEnable (CE)	CAM Enable When this field is cleared to 0, the CAM is disabled and no address comparisons are performed on received frames. Also, this field must be cleared to 0 when performing read or write operations on the CAM. Setting this field to 1 enables the CAM. For predictable operation, this field should be set before enabling the MAC.	R/W	0

ETHERNET CONTROL REGISTER (ENETCONTROL, OFFSET 0X42C)

The four-byte Ethernet Control register is both readable and writable and is initialized to 0x0 at device reset.

Table 99: Ethernet Control Register Bit Field Descriptions

Field	Name	Description	Access	Default
31:4	Reserved	–	R/W	0
3	ExtPHYSelect (EP)	External PHY Select Writing a 1 to this field turns on the external MII interface and disable internal Ethernet transceiver. This field is 0 after reset. In devices without internal Ethernet PHYs, software must set this field to one before frames can be transmitted or received.	R/W	0
2	EMACSoftReset (ES)	EMAC Soft Reset Reset the Ethernet MAC by setting this field and then wait for it to be cleared by the internal logic before accessing any of the MAC registers. The soft reset resets all the logic in the Ethernet MAC. This field is 0 after reset.	R/W	0
1	EMACDisable (ED)	EMAC Disable Disable the Ethernet MAC by writing a 1 to this field and then wait for it to be cleared by the internal logic. When the field is cleared to 0 by the internal logic that means the MAC is disabled and it is okay to update registers that should only be updated when the MAC is disabled. Writing a 0 to this field does nothing. Setting this field automatically clears the EMACEnable field. This field is 0 after reset.	R/W	0
0	EMACEnable (EE)	EMAC Enable Enable the Ethernet MAC by writing a 1 to this field. Writing a 0 to this field does nothing. This field is cleared by writing a 1 to the EMACDisable field. This field is cleared to 0 after reset. Do not update MAC registers when the MAC is enabled to avoid unknown behavior. Disable the Ethernet MAC first before updating any registers.	R/W	0

TRANSMIT CONTROL REGISTER (XMTCONTROL, OFFSET 0X430)

The four-byte Transmit Control register is both readable and writable and is initialized to 0x0 at device reset.

Table 100: Transmit Control Register Bit Field Descriptions

Field	*Name*	*Description*	*Access*	*Default*
31:4	Reserved	–	R/W	0
3	SmSlotTime (SS)	Small Slot Time When this field is cleared to 0, the slot time (512 bit times) used to calculate backoff delays is set to 512 bit times. When the field is set to 1, the backoff delay is set to 1 bit time. This field should only be used for testing.	R/W	0
2	SingleBackoffEn (SB)	Single Backoff Enable When this field is cleared to 0, the random backoff algorithm is enabled. When this field is set to 1, the random backoff algorithm is disabled. This field can be set for full-duplex operation. This field should not be set for half-duplex operation.	R/W	0
1	FLowMode (FM)	Flow Mode The transmitter supports two modes of initiating pause frame transmission with the FLOWMODE control signal. When FLOWMODE = 0, the device complies with the 802.3x requirement to defer flow control frame transmission until any frame with transmission in progress completes. Setting FLOWMODE = 1 allows the device to abort transmit frames in progress that are still within the collision window and subsequently send the flow control frame. The aborted frame is retransmitted after the flow control frame. In either mode, when the condition that caused the flow control state to be entered is no longer present, a MAC control pause frame is sent with the Pause_Time field set to 0.	R/W	0
0	FullDuplex (FD)	Full-Duplex Set this field to 1 to put the Ethernet MAC into full-duplex mode. Otherwise, if this field is cleared to 0, the MAC operates in half-duplex mode. This field is cleared to 0 after reset.	R/W	0

TRANSMIT WATERMARK REGISTER (XMTWATERMARK, OFFSET 0X434)

The four-byte Transmit Watermark register is both readable and writable and is initialized to 0x10 at device reset.

Table 101: Transmit Watermark Register Bit Field Descriptions

Field	Name	Description	Access	Default
31:6	Reserved	–	R/W	–
5:0	XmtWatermark[5:0] (WM)	Transmit Watermark [6] This value represents the number of 32-bit words that are stored in the transmit FIFO before it begins transmitting a frame. This register should only be updated while the transmitter is disabled. A value of 0 is illegal. The recommended value for a driver to use for the XmtWatermark field is 56 (or 224 bytes). If the DMA engine forwards a complete frame that is smaller than 224 bytes, for example a 64 byte frame, then the transmitter will ignore the XmtWatermark value and transmit the frame onto the wire regardless. It is possible to put the Ethernet transmitter in a locked state if the ISBMaxBlen[4:0] field in word 0x11 of the SPROM for the transmit DMA channel and the XmtWatermark register are not programmed correctly. The watermark must be small enough that a maximum number of bursts into the FIFO before filling it exceeds the programmed watermark. An equation to use to avoid this condition is as follows: XMIT_WATERMARK = 63 - (63% ISBMaxBlen) This insures that no matter what size the watermark is set to a discrete number of bursts exceeds the watermark without exceeding the 64 word size of the FIFO.	R/W	0x10

MIB CONTROL REGISTER (MIBCONTROL, OFFSET 0X438)

The four-byte MIB Control register is both readable and writable and is initialized to 0x0 at device reset.

Table 102: MIB Control Register Bit Field Descriptions 7

Field	*Name*	*Description*	*Access*	*Default*
31:1	Reserved	–	R/W	0
0	RO (RO)	Read Only When this field is set to 1, a read to a MIB statistic counter causes that counter to be reset to 0. A value of 0 in this field disables this feature.	R/W	0

MANAGEMENT INFORMATION BLOCK (MIB) STATISTIC COUNTERS

The Ethernet Management Information Block contains a set of 47 statistics counters that support the following specifications:

- IETF RFC 1757 (RMON statistics group)
- IETF RFC 1643 (Ethernet-like MIB)
- Ethernet MIB (IEEE 802.3u)

The MIB logic utilizes an SRAM array, individual holding registers, and a centralized adder to maintain the individual statistics. The current value of the statistics are individually passed through the adder during update. All current values of 16 bits or less are maintained in the SRAM, while the larger 32-bit values are maintained in individual registers. Note that some statistic definitions overlap between specifications and two statistics can occupy the same physical address. In other words, they are the same statistic value contained in two different specifications. The statistics counters are listed in Table 103.

- **Counter Reset**—All counters must be reset to 0 by user software.
- **Counter Reads**—When a counter is read, its count is temporarily frozen while the read is in progress. Any requested increments to the counter are delayed until the read of the counter is complete.
- **Counter Reset On Read**—Normally, the counter value is unaffected when a read is performed. By setting the RO field in the MIBControl register, a read to a counter resets its value to 0.

Table 103: MIB Statistics Counters

Counter	*Description*	*Offset*	*Size*	*R/W*	*Reset Value*
Tx_gd_octets	The total number of octets transmitted in good packets	0x500	4	R/W	0x0
Tx_gd_pkts	The total number of frames that are successfully transmitted	0x504	2	R/W	0x0
Tx_all_octets	All octets transmitted to the network. (Excludes preamble and SFD bits, includes FCS octets)	0x508	4	R/W	0x0
Tx_all_pkts	Good or bad packets transmitted.	0x50C	2	R/W	0x0
Tx_brdcast	Good broadcast packets transmitted, not including multicast packets	0x510	2	R/W	0x0
Tx_mult	Multicast packets transmitted with a good CRC.	0x514	2	R/W	0x0
Tx_64	Good or bad transmitted packets 64 bytes in length (Excluding preamble)	0x518	2	R/W	0x0
Tx_65_127	Good or bad transmitted packets between 65 and 127 bytes, inclusive, in length (Excluding preamble)	0x51C	2	R/W	0x0
Tx_128_255	Good or bad transmitted packets between 128 and 255 bytes, inclusive, in length (Excluding preamble)	0x520	2	R/W	0x0
Tx_256_511	Good or bad transmitted packets between 256 and 511 bytes, inclusive, in length (Excluding preamble)	0x524	2	R/W	0x0
Tx_512_1023	Good or bad transmitted packets between 512 and 1023 bytes, inclusive, in length (Excluding preamble)	0x528	2	R/W	0x0
Tx_1024_max	Good or bad transmitted packets between 1024 and RX_LENGTH bytes, inclusive, in length (Excluding preamble)	0x52C	2	R/W	0x0
Tx_jab	Transmitted packets with length > TX_MAX_LEN and a bad CRC.	0x530	2	R/W	0x0
Tx_ovr	Transmitted packets with length > TX_MAX_LEN and a good CRC	0x534	2	R/W	0x0

Table 103: MIB Statistics Counters (Cont.)

Counter	*Description*	*Offset*	*Size*	*R/W*	*Reset Value*
Tx_frag	Transmitted packets with length < 64 and a bad CRC	0x538	2	R/W	0x0
Tx_underrun	Packets with a transmit FIFO underflow	0x53C	2	R/W	0x0
Tx_col	The total number of collisions seen by this transmitter	0x540	2	R/W	0x0
Tx_1_col	Packets transmitted successfully with only 1 collision	0x544	2	R/W	0x0
Tx_m_col	Packets transmitted with more than 1 collision	0x548	2	R/W	0x0
Tx_ex_col	Transmitted packets that attempted more than 15 retries and failed	0x54C	2	R/W	0x0
Tx_late	Number of times a collision occurred after 512 bit times	0x550	2	R/W	0x0
Tx_def	Packet whose transmission is delayed due to a busy network.	0x554	2	R/W	0x0
Tx_crs	Number of times CRS is lost or never asserted when transmitting a frame.	0x558	2	R/W	0x0
Tx_paus	MAC control frames with Pause opcode transmitted.	0x55c	2	R/W	0x0
Rx_gd_octets	The total number of octets received in good packets	0x580	4	R/W	0x0
Rx_gd_pkts	The total number of frames that are successfully received (includes frames that are normally dropped when the adapter is not operating in "promiscuous" mode).	0x584	2	R/W	0x0
Rx_all_octets	Octets received from the network, including those in bad packets. (Excludes preamble and SFD bits, includes FCS octets)	0x588	4	R/W	0x0
Rx_all_pkts	Good or bad packets received.	0x58C	2	R/W	0x0
Rx_brdcast	Broadcast packets received with a good CRC. (Excludes multicast packets)	0x590	2	R/W	0x0
Rx_mult	Multicast packets received with a good CRC.	0x594	2	R/W	0x0
Rx_64	Good or bad received packets 64 bytes in length (Excluding preamble)	0x598	2	R/W	0x0
Rx_65_127	Good or bad received packets between 65 and 127 bytes, inclusive, in length (Excluding preamble)	0x59C	2	R/W	0x0
Rx_128_255	Good or bad received packets between 128 and 255 bytes, inclusive, in length (Excluding preamble)	0x5A0	2	R/W	0x0
Rx_256_511	Good or bad received packets between 256 and 511 bytes, inclusive, in length (Excluding preamble)	0x5A4	2	R/W	0x0
Rx_512_1023	Good or bad received packets between 512 and 1023 bytes, inclusive, in length (Excluding preamble)	0x5A8	2	R/W	0x0
Rx_1024_max	Good or bad received packets between 1024 and RX_LENGTH bytes, inclusive, in length (Excluding preamble)	0x5AC	2	R/W	0x0
Rx_jab	Received packets with length > 1536 and a bad CRC.	0x5B0	2	R/W	0x0
Rx_ovr	Received packets with length > RX_LENGTH and a good CRC.	0x5B4	2	R/W	0x0
Rx_frag	Received packets with length < 64 and a bad CRC.	0x5B8	2	R/W	0x0
Rx_drop	The total number of packets received that experienced an overrun.	0x5BC	2	R/W	0x0
Rx_CRC_align	Received packets with a bad CRC or odd number of nibbles.	0x5C0	2	R/W	0x0
Rx_und	Received packets with length < 64 with a good CRC.	0x5C4	2	R/W	0x0

Table 103: MIB Statistics Counters (Cont.)

Counter	Description	Offset	Size	R/W	Reset Value
Rx_CRC	The total number of received frames that are an integral number of bytes and do not pass the FCS check.	0x5C8	2	R/W	0x0
Rx_align	The total number of received frames that are not an integral number of bytes and do not pass the FCS check.	0x5CC	2	R/W	0x0
Rx_sym	RXER occurs while CRS is asserted.	0x5D0	2	R/W	0x0
Rx_paus	MAC control frames with Pause opcode received.	0x5D4	2	R/W	0x0
Rx_cntrl	MAC control frames without Pause opcode received.	0x5D8	2	R/W	0x0

ETHERNET PHY REGISTERS

The Ethernet PHY register include the following:

MII CONTROL REGISTER (MIICONTROL, OFFSET 0X00)

The two-byte MII Control register is both readable and writable and is initialized to 0x3000 at device reset.

Table 104: MII Control Register Bit Field Descriptions

Field	Name	Description	Access	Default
15	Reset (RS)	Reset To reset the device by software control, this field must be set to 1 using an MII write operation. After the reset process is complete, this field is cleared to 0. Writes to other fields have no effect until the reset process is completed (in approximately 1 µs). Writing a 0 to this field has no effect. Because this field is self-clearing after a write operation, it returns a 0 when read.	R/W	0
14	Loopback (LB)	Loopback The device can be placed into loopback mode by writing a 1 to this field. Clear the loopback mode by writing a 0 to this field, or by resetting the chip. When this field is read, it returns a 1 when the chip is in loopback mode, otherwise it returns a 0.	R/W	0
13	ForcedSpeedSel (FS)	Forced Speed Selection If auto-negotiation is enabled (both auto-negotiation pin and field are enabled) or disabled by hardware control (auto-negotiation pin is pulled low), this field has no effect on the speed selection. However, if auto-negotiation is disabled by software control, the operating speed of the device can be forced by writing the appropriate value to this field. Writing a 1 to this field forces 100BASE-X operation, while writing a 0 forces 10BASE-T operation. When this field is read, it returns the value of the software-controlled forced speed selection only. To read the overall state of forced speed selection, including both hardware and software control, use the Force100/10 field of the AuxCtlStatus register (see "100BASE-X Auxiliary Control/Status Register (AuxCtlStatus, offset 0x18)" on page 114).	R/W	1
12	AutoNegEn (AE)	Auto-Negotiation Enable Auto-negotiation can be disabled by one of two methods: hardware or software control. If the Auto-Negotiation input from the MAC is driven to a logic 0, auto-negotiation is disabled by hardware control. If this field is cleared to 0, auto-negotiation is disabled by software control. When auto-negotiation is disabled in this manner, writing a 1 to this field or resetting the chip re-enables auto-negotiation. Writing to this field has no effect when auto-negotiation has been disabled by hardware control. When read, this field returns the value most recently written to this location, or 1 if it has not been written since the last chip reset. ***Note:*** Disabling auto-negotiation also disables auto-MDIX	R/W	1
11	PowerDown (PD)	Power Down The power modes of the device are not accessible by this MII register field. Use the Low Power Mode Enable input from the MAC instead.	R/W	0

Table 104: MII Control Register Bit Field Descriptions (Cont.)

Field	Name	Description	Access	Default
10	Isolate (IS)	Isolate The PHY can be isolated from its Media Independent Interface by writing a 1 to this field. All MII outputs are tri-stated and all MII inputs are ignored. Because the MII management interface is still active, isolate mode can be cleared by writing a 0 to this field, or by resetting the chip. When this field is read, it returns a 1 when the chip is in isolate mode; otherwise it returns a 0.	R/W	0
9	RestartAutoNeg (RA)	Restart Auto-Negotiation This self-clearing field allows the auto-negotiation process to be restarted, regardless of the current status of the auto-negotiation state machine. For this field to have an effect, auto-negotiation must be enabled. Writing a 1 to this field restarts the auto-negotiation; writing a 0 to this field has no effect. Because the field is self-clearing after only a few cycles, it always returns a 0 when read. The operation of this field is identical to the RestartAutoNeg field of the AuxMultiPHY register (see "Auxiliary Multiple PHY Register (AuxMultiPHY, offset 0x1E)" on page 124).	R/W	0
8	DuplexMode (DM)	Duplex Mode By default, the device powers up in half-duplex mode. The device can be forced into full-duplex mode by writing a 1 to this field while auto-negotiation is disabled. Half-duplex mode can be resumed by writing a 0 to this field or by resetting the chip.	R/W	0
7	CollisionTest (CT)	Collision Test Test the Collision Detect output by activating the collision test mode. While in this mode, asserting the MII Transmit Enable input from the MAC causes the Collision Detect output to go high within 512 bit times. Deasserting the MII Transmit Enable input from the MAC causes the Collision Detect output to go low within 4 bit times. Writing a 1 to this field enables the collision test mode. Writing a 0 to this field or resetting the chip disables the collision test mode. When this field is read, it returns a 1 when the collision test mode has been enabled; otherwise it returns a 0. This field should only be set while in loopback test mode.	R/W	0
6:0	Reserved	–	R/W	0

MII STATUS REGISTER (MIISTATUS, OFFSET 0X01)

The two-byte MII Status register is both readable and writable and is initialized to 0x7809 at device reset.

Table 105: MII Status Register Bit Field Descriptions

Field	Name	Description	Access	Default
15	Reserved	–	RO	1
14	100BaseTFullDuplex (CF)	100BASE-X Full-Duplex Capability The device is capable of 100BASE-X full-duplex operation, and returns a 1 when this field is read.	RO	1
13	100BaseTHalfDuplex (CH)	100BASE-X Half-Duplex Capability The device is capable of 100BASE-X half-duplex operation, and returns a 1 when this field is read.	RO	0
12	10BaseTFullDuplex (XF)	10BASE-T Full-Duplex Capability The device is capable of 10BASE-T full-duplex operation, and returns a 1 when this field is read.	RO	1
11	10BaseTHalfDuplex (XH)	10BASE-T Half-Duplex Capability The device is capable of 10BASE-T half-duplex operation, and returns a 1 when this field is read.	RO	1
10:7	Reserved	–	RO	0
6	PreambleSuppress (PS)	Preamble Suppression This field is the only writable field in the Status Register. Setting this field to 1 allows subsequent MII management frames to be accepted with or without the standard preamble pattern. When preamble suppression is enabled, only two preamble bits are required between successive management commands, instead of the normal 32.	R/W	0
5	AutoNegComplete (CM)	Auto-Negotiation Complete Returns a 1 if auto-negotiation process has been completed and the contents of the AutoNegAdvertise, LinkPartnerAbility, and AutoNegExpansion registers are valid.	RO	0
4	LPRemoteFault (RM)	Link Partner Remote Fault The PHY returns a 1 in this field when its link partner has signalled a far-end fault condition. When a far-end fault occurs, the field is latched at 1 and remains so until the register is read and the remote fault condition has been cleared.	RO	0
3	AutoNegCapability (AC)	Auto-Negotiation Capability The device can perform IEEE auto-negotiation, and returns a 1 when the RemoteFault field is read, regardless of whether the auto-negotiation function has been disabled.	RO	1
2	LinkStatus (LS)	Link Status The device returns a 1 on this field when the link state machine is in the link pass state, indicating that a valid link has been established. Otherwise, it returns 0. When a link failure occurs after the link pass state has been entered, this field is etched at 0 and remains so until the field is read. After the field is read, it becomes 1 when the link pass state is entered again.	RO	0

Table 105: MII Status Register Bit Field Descriptions (Cont.)

Field	*Name*	*Description*	*Access*	*Default*
1	JabberDetect (JD)	Jabber Detect 10BASE-T operation only. The device returns a 1 on this field if a jabber condition has been detected. After this field is read once, or if the chip is reset, it reverts to 0.	RO	0
0	ExtCapability (EC)	Extended Capability The device supports extended capability registers, and returns a 1 when this field is read. Several extended registers have been implemented in the device, and their field functions are defined later in this section.	RO	1

PHY IDENTIFIER HIGH REGISTER (PHYIDHIGH, OFFSET 0X02)

The two-byte PHY Identifier High register is both readable and writable and is initialized to 0x0040 at device reset.

Table 106: PHY Identifier High Register Bit Field Descriptions

Field	Name	Description	Access	Default
15:0	MIIAddr2 (A2)	MII Address 02 This field contains the upper part of the Broadcom Organizationally Unique Identifier (OUI) issued by the IEEE. It is a 24-bit number, 00-10-18, expressed as hex values. This number also contains the Broadcom Model number and Revision number for the BCM440X part. The translation for the PHYIDHigh portion of the number is PHYIDHigh[15:0] = OUI[21:6]. The two most significant bits of the OUI are not represented (OUI[23:22]).	R/W	0x0040

PHY IDENTIFIER LOW REGISTER (PHYIDLOW, OFFSET 0X03)

The two-byte PHY Identifier Low register is both readable and writable and is initialized to 0x6000 at device reset.

Table 107: PHY Identifier Low Register Bit Field Descriptions

Field	Name	Description	Access	Default
15:0	MIIADDR3 (A3)	MII ADDRESS 03 This field contains the lower part of the Broadcom Organizationally Unique Identifier (OUI) issued by the IEEE. This number also contains the Broadcom Model number and Revision number for the BCM440X part. The translation for the PHYIDLow portion of the number is PHYLow[15:0] = OUI[5:0] + MODEL[5:0] + REV[3:0].	R/W	0x6000

AUTO-NEGOTIATION ADVERTISEMENT REGISTER (AUTONEGADVERTISE, OFFSET 0X04)

The two-byte Auto-Negotiation Advertisement register is both readable and writable and is initialized to 0x01E1 at device reset.

Table 108: Auto-Negotiation Advertisement Register Bit Field Descriptions

Field	*Name*	*Description*	*Access*	*Default*
15	NextPage (NP)	Next Page. The device supports the next page function.	R/W	0
14	Reserved	–	R/W	0
13	LPRemoteFault (RF)	Link Partner Remote Fault. Writing a 1 to this field causes a remote fault indicator to be sent to the link partner during auto-negotiation. Writing a 0 to this field or resetting the chip clears the remote fault transmission bit. This field returns the value last written to it, or else 0 if no write has been completed since the last chip reset.	R/W	0
12:11	Reserved	–	R/W	0
10	Pause (PS)	Pause Pause operation for full-duplex links. The use of this field is independent of the negotiated data rate, medium, or link technology. The setting of this field indicates the availability of additional DTE capability when full-duplex operation is in use. This field is used by the MAC to communicate pause capability to its link partner and has no effect on PHY operation.	R/W	0
9:5	Advertisement (AD)	Advertisement Use this field to customize the ability information transmitted to the link partner. The default value for each bit reflects the abilities of the device. By writing a 1 to any of the bits, the corresponding ability can be transmitted to the link partner. Writing a 0 to any bit causes the corresponding ability to be suppressed from transmission. Resetting the chip restores the default bit values. Reading the register returns the values last written to the corresponding bits, or else the default values if no write has been completed since the last chip reset.	R/W	01111b
4:0	Selector (SL)	Selector This field contains the value 00001, indicating that the chip belongs to the 802.3 class of PHY transceivers.	R/W	1

AUTO-NEGOTIATION LINK PARTNER ABILITY REGISTER (LINKPARTNERABILITY, OFFSET 0X05)

The two-byte Auto-Negotiation Link Partner Ability register is both readable and writable and is initialized to 0x0000 at device reset.

Table 109: Auto-Negotiation Link Partner Ability Register Bit Field Descriptions

Field	Name	Description	Access	Default
15	LPNextPage (NP)	Link Partner Next Page This field returns a value of 1 when the link partner implements the next page function and has next page information that it wants to transmit. The device does not implement the next page function and ignores the next page bit, except to copy it to this register.	R/W	0
14	LPAck (AK)	Link Partner Acknowledge This field is used by auto-negotiation to indicate that a device has successfully received its link partner's link code word.	R/W	0
13	LPRemoteFault (RM)	Link Partner Remote Fault This field returns a value of 1 when the link partner signals that a remote fault has occurred. The device simply copies the value to this register and does not act upon it.	R/W	0
12:11	Reserved	–	R/W	0
10	LPAdvertisePause (AP)	Link Partner Advertise Pause This field indicates that the Pause field is set.	R/W	0
9:5	LPAdvertise (LA)	Link Partner Advertise This field reflects the abilities of the link partner. A 1 on any of the bits in this field indicates that the link partner is capable of performing the corresponding mode of operation. These bits are cleared to 0 when auto-negotiation is restarted or when the chip is reset.	R/W	0
4:0	LPSelector (LS)	Link Partner Selector This field reflects the value of the link partner's selector field. The bits in this field are cleared to 0 when auto-negotiation is restarted or when the chip is reset.	R/W	0

AUTO-NEGOTIATION EXPANSION REGISTER (AUTONEGEXPANSION, OFFSET 0X06)

The two-byte Auto-Negotiation Expansion register is both readable and writable and is initialized to 0x0004 at device reset.

Table 110: Auto-Negotiation Expansion Register Bit Field Descriptions

Field	*Name*	*Description*	*Access*	*Default*
15:5	Reserved	–	R/W	0
4	AutoNegParallel DetFlt (PF)	Parallel Detection Fault This read-only field gets latched high when a parallel detection fault occurs in the auto-negotiation state machine. For further details, refer to the IEEE standard. This field is cleared to 0 after the register is read, or when the chip is reset.	R/W	0
3	LPNextPageAble (NA)	Link Partner Next Page Able This field returns a 1 when the link partner has next page capabilities. It has the same value as the LPNextPage field of the LinkPartnerAbility register.	R/W	0
2	NextPage (NP)	Next Page Able The device returns 1 when this field is read, indicating that it has next page capabilities.	R/W	1
1	LPPageRcv (PR)	Link Partner Page Received This field is latched high when a new link code word is received from the link partner, checked, and acknowledged. It remains high until the register is read, or until the chip is reset.	R/W	0
0	LPAutoNegAble (LP)	Link Partner Auto-Negotiation Able This field returns a 1 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 value of 0.	R/W	0

AUTO-NEGOTIATION NEXT PAGE TRANSMIT REGISTER (NEXTPAGE, OFFSET 0X07)

The two-byte Auto-Negotiation Next Page Transmit register is both readable and writable and is initialized to 0x2001 at device reset.

Table 111: Auto-Negotiation Next Page Transmit Register Bit Field Descriptions

Field	Name	Description	Access	Default
15	NextPage (NP)	Next Page This field, when set to 1, indicates that additional pages will follow. When cleared to 0, this field indicates that this is the last page to be transmitted.	R/W	0
14	Reserved	–	R/W	0
13	MsgPage (MP)	Message Page This field differentiates a Message Page from an Unformatted Page. When set to 1, it indicates a Message Page; when cleared to 0, it indicates an unformatted page.	R/W	1
12	Acknowledge2 (A2)	Acknowledge 2 This field, when set to 1, indicates that a device can comply with the message.	R/W	0
11	Toggle (TG)	Toggle This field is used by the Arbitration function to ensure synchronization with the link partner during next page exchange. When set to 1, it indicates that the previous value of the transmitted link code word equalled logic 0. When cleared to 0, it indicates that the previous value of the transmitted link code word equalled logic 1.	R/W	0
10:0	Msg/UnformatCode (MU)	Message Code/Unformatted Code Message Code is an 11-bit field that encodes 2048 possible messages. Unformatted Code is an 11-bit field that can contain an arbitrary value.	R/W	0

AUTO-NEGOTIATION LINK PARTNER NEXT PAGE TRANSMIT REGISTER (LPNEXTPAGE, OFFSET 0X08)

The two-byte Auto-Negotiation Link Partner Next Page Transmit register is both readable and writable and is initialized to 0x0000 at device reset.

Table 112: Auto-Negotiation Link Partner Next Page Transmit Register Bit Field Descriptions

Field	*Name*	*Description*	*Access*	*Default*
15	NextPage (NP)	Next Page This field, when set to 1, indicates that additional pages will follow. When cleared to 0, this field indicates that this is the last page to be transmitted.	R/W	0
14	Reserved	–	R/W	0
13	MsgPage (MP)	Message Page This field differentiates a Message Page from an Unformatted Page. When set to 1, it indicates a Message Page. When cleared to 0, it indicates an unformatted page.	R/W	0
12	Acknowledge2 (A2)	Acknowledge 2 This field, when set to 1, indicates that a device can comply with the message.	R/W	0
11	Toggle (TG)	Toggle This field is used by the Arbitration function to ensure synchronization with the link partner during next page exchange. When set to 1, it indicates that the previous value of the transmitted link code word equalled logic 0. When cleared to 0, it indicates that the previous value of the transmitted link code word equalled logic 1.	R/W	0
10:0	Msg/UnformatCode (MU)	Message Code/Unformatted Code Message Code is an 11-bit field that encodes 2048 possible messages. Unformatted Code is an 11-bit field that can contain an arbitrary value.	R/W	0

100BASE-X AUXILIARY CONTROL REGISTER (100BASE-XAUXCONTROL, OFFSET 0X10)

The two-byte 100BASE-X Auxiliary Control register is both readable and writable and is initialized to 0x0000 at device reset.

Table 113: 100BASE-X Auxiliary Control Register Bit Field Descriptions

Field	Name	Description	Access	Default
15:14	Reserved	–	R/W	0
13	XmtDisable (XD)	Transmit Disable The transmitter can be disabled by writing a 1 to this field. The transmitter output (TD±) is forced into a high impedance state.	R/W	0
12:11	Reserved	–	R/W	0
10	Bypass4B5BEncDec (ED)	Bypass 4B5B Encoder/Decoder The 4B5B encoder and decoder can be bypassed by writing a 1 to this field. The transmitter sends 5B codes from the MII Transmit Error, Transmit Data [1], and Transmit Data [0] inputs from the MAC directly to the scrambler. The MII Transmit Enable input from the MAC must be active and frame encapsulation (insertion of J/K and T/R codes) is not performed. The receiver places descrambled and aligned 5B codes onto the MII Receive Error, Receive Data [1], and Receive Data [0] outputs. The MII Carrier Sense output can be asserted when a valid frame is received.	R/W	0
9	BypassScrambler Descrambler (SD)	Bypass Scrambler/Descrambler The stream cipher function can be disabled by writing a 1 to this field. The stream cipher function is re-enabled by writing a 0 to this field.	R/W	0
8	BypassNRZI (BN)	Bypass NRZI Encoder/Decoder The NRZI encoder and decoder can be bypassed by writing a 1 to this field, causing three-level NRZ data to be transmitted and received on the cable. Normal operation (three-level NRZI encoding and decoding) can be re-enabled by writing a 0 to this field.	R/W	0
7	BypassRcvSym (BR)	Bypass Receive Symbol Alignment Receive symbol alignment can be bypassed by writing a 1 to this field. When used in conjunction with the Bypass4B5BEncDec field, unaligned 5B codes are placed directly on the MII Receive Error, Receive Data [1], and Receive Data [0] outputs.	R/W	0
6	BaselineWanderDis (BW)	Baseline Wander Correction Disable The baseline wander correction circuit can be disabled by writing a 1 to this field. The device corrects for baseline wander on the receive data signal when this field is cleared to 0.	R/W	0
5	FEFEn (FF)	FEF Enable Controls the Far End Fault (FEF) mechanism associated with 100BASE-FX operation. A 1 enables the FEF function and a 0 disables it.	R/W	0
4:3	Reserved	–	R/W	0
2	ExtRMIIFIFOEn (ER)	Extended RMII FIFO Enable Controls the extended RMII FIFO mechanism.	R/W	0

Table 113: 100BASE-X Auxiliary Control Register Bit Field Descriptions (Cont.)

Field	*Name*	*Description*	*Access*	*Default*
1	RMIIOBEn (RM)	RMII Out-of-Band Enable Controls the RMII out-of band mechanism within the RMII receive logic. When this field is set to 1, RMII Out-of-Band is enabled. When it is cleared to 0, RMII Out-of-Band is disabled.	R/W	0
0	Reserved	–	R/W	0

100BASE-X AUXILIARY STATUS REGISTER (100BASE-XAUXSTATUS, OFFSET 0X11)

The two-byte 100BASE-X Auxiliary Status register is both readable and writable and is initialized to 0x0000 at device reset.

Table 114: 100BASE-X Auxiliary Status Register Bit Field Descriptions

Field	*Name*	*Description*	*Access*	*Default*
15:12	Reserved	–	R/W	0
11	RMIIOverUnderErr (OU)	RMII Overrun/Underrun Error The PHY returns a 1 in this field, when the RMII receive FIFO encounters an over-run or under-run condition.	R/W	0
10	FXMode (FX)	FX Mode Returns a value derived from the 100BASE-FX Signal Detect (SD±) input from the MAC. Returns 1 when SD+ are driven with a valid differential signal level. Returns 0 when both SD+ and SD- are simultaneously driven low.	R/W	0
9	Locked (LK)	Locked The PHY returns a 1 in bit 9 when the descrambler is locked to the incoming data stream. Otherwise it returns a 0.	R/W	0
8	Current100Base-X LinkStatus (LS)	Current 100BASE-X Link Status The PHY returns a 1 in this field when the 100BASE-X link status is good. Otherwise it returns a 0.	R/W	0
7	RemoteFault (RM)	Remote Fault The PHY returns a 1 while its link partner is signalling a far-end fault condition. Otherwise it returns a 0.	R/W	0
6	Reserved	–	R/W	0
5	FalseCarrier (FC)	False Carrier Detected The PHY returns a 1 in this field of the Extended Status Register if a false carrier has been detected since the last time this register was read. Otherwise it returns a 0.	R/W	0
4	BadESD (BE)	Bad ESD Detected The PHY returns a 1 in this field if an end of stream delimiter error has been detected since the last time this register was read. Otherwise it returns a 0.	R/W	0
3	RcvError (RE)	Receive Error Detected The PHY returns a 1 in this field if a packet was received with an invalid code since the last time this register was read. Otherwise it returns a 0.	R/W	0
2	XmtError (XE)	Transmit Error Detected The PHY returns a 1 in this field if a packet was received with a transmit error code since the last time this register was read. Otherwise it returns a 0.	R/W	0

Table 114: 100BASE-X Auxiliary Status Register Bit Field Descriptions (Cont.)

Field	*Name*	*Description*	*Access*	*Default*
1	LockError (LE)	Lock Error Detected The PHY returns a 1 in this field if the descrambler has lost lock since the last time this register was read. Otherwise it returns a 0.	R/W	0
0	MLT3CodeError (ME)	MLT3 Code Error Detected The PHY returns a 1 in this field 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.	R/W	0

100BASE-X Receive Error Counter (100BASE-XRcvErrorCtr, offset 0x12)

The two-byte 100BASE-X Receive Error counter is both readable and writable and is initialized to 0x0000 at device reset.

Table 115: 100BASE-X Receive Error Counter Bit Field Descriptions

Field	Name	Description	Access	Default
15:0	RcvErrCtr (RE)	Receive Error Counter [15:0] This counter increments each time the device receives a non-collision packet containing at least one receive error. The counter automatically clears itself when read. When the counter reaches its maximum value, 0xFF, it stops counting receive errors until cleared.	R/W	0x0000

100BASE-X False Carrier Sense Counter (100BASE-XCarrierSenseCtr, offset 0x13)

The two-byte 100BASE-X Carrier Sense counter is both readable and writable and is initialized to 0x0000 at device reset.

Table 116: 100BASE-X False Carrier Sense Counter Bit Field Descriptions

Field	Name	Description	Access	Default
15:8	DMIIOverUnderCtr (OU)	RMII Overrun/Underrun Counter The RMII over-run/under-run counter increments each time the device detects an over-run or under-run of the RMII FIFOs. The counter automatically clears itself when read. When the counter reaches its maximum value, 0xFF, it stops counting over-run/ under-run errors until cleared.	R/W	0
7:0	FalseCSCtr (FC)	False Carrier Sense Counter This counter increments each time the device detects a false carrier on the receive input. This counter automatically clears itself when read. When the counter reaches its maximum value, 0xFF, it stops counting False Carrier Sense Errors until cleared.	R/W	0

100BASE-X DISCONNECT COUNTER (100BASE-XDISCONNECTCTR, OFFSET 0X14)

The two-byte 100BASE-X Disconnect counter is both readable and writable and is initialized to 0x0200 at device reset.

Table 117: 100BASE-X Disconnect Counter Bit Field Descriptions

Field	*Name*	*Description*	*Access*	*Default*
15	RMIIFastRXD (RF)	RMII Fast RXD Extended FIFO operation only. This field indicates the FIFO state machine has detected fast receive data relative to the 25MHz Reference Clock input.	R/W	–
14	RMIISlowRXD (RS)	RMII Slow RXD Extended FIFO operation only. This field indicates the FIFO state machine has detected slow receive data relative to the 25MHz Reference Clock input.	R/W	–
13:0	Reserved	–	R/W	–

100BASE-X AUXILIARY CONTROL/STATUS REGISTER (AUXCTLSTATUS, OFFSET 0X18)

The two-byte Auxiliary Control/Status register is both readable and writable and is initialized to 0x003x at device reset.

Table 118: 100BASE-X Auxiliary Control/Status Register Bit Field Descriptions

Field	Name	Description	Access	Default
15	JabberDisable (JD)	Jabber Disable 10BASE-T operation only. This field 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. By writing a 1 to this field, the jabber detect function is disabled. Writing a 0 to this field or resetting the chip restores normal operation. Reading this field returns the value of jabber detect disable.	R/W	–
14	LinkForce (LF)	Link Force Writing a 1 to this field allows the user to disable the Link Integrity state machines, and place the device into forced link pass status. Writing a 0 to this field or resetting the chip restores the Link Integrity functions. Reading this field returns the value of the force link bit.	R/W	–
13:8	Reserved	–	R/W	–
7:6	HSQLSQ (HL)	HSQ and LSQ Extend or decrease 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. Extending the squelch levels allows the device to operate properly over longer cable lengths. Decreasing the squelch levels can be useful in situations where there is a high level of noise present on the cables. Reading this field returns the value of the squelch levels.	R/W	–
5:4	EdgeRate (ER)	Edge Rate This field contains control bits used to program the transmit DAC output edge rate in 100BASE-TX mode. A larger value in this field produces slower transitions on the transmit waveform.	R/W	–
3	AutoNegIndicat (AN)	Auto-Negotiation Indication This read-only field indicates whether auto-negotiation has been enabled or disabled on the device. A combination of a 1 in the AutoNegEn field of the MIIControl register and a logic 1 on the Auto-Negotiation input from the MAC is required to enable auto-negotiation. When auto-negotiation is disabled, this field returns a 0. At all other times, it returns a 1.	RO	–

Table 118: 100BASE-X Auxiliary Control/Status Register Bit Field Descriptions (Cont.)

Field	*Name*	*Description*	*Access*	*Default*
2	Force100/10Indicat (FI)	Force100/10 Indication This read-only field returns a value of 0 when one of following cases is true: The Auto-Negotiation input from the MAC is low AND the F100 pin is low or the Auto-Negotiation input from the MAC is high AND the AutoNegEn field of the MIIControl register has been written 0 and the ForcedSpeedSel field of the MIIControl register has been written 0. When this field is 0, the speed of the chip is 10BASE-T. In all other cases, either the speed is not forced (auto-negotiation is enabled), or the speed is forced to 100BASE-X.	RO	–
1	SpeedIndication (SI)	Speed Indication This read-only field shows the true current operation speed of the device. A 1 indicates 100BASE-X operation, and a 0 indicates 10BASE-T. While the auto-negotiation exchange is performed, the device is always operating at 10BASE-T speed.	RO	–
0	Full-DuplexIndicat (FD)	Full-Duplex Indication This read-only field returns a 1 when the device is in full-duplex mode. In all other modes, it returns a 0.	RO	–

100BASE-X AUXILIARY STATUS SUMMARY REGISTER (AUXSTATUSSUMMARY, OFFSET 0X19)

The two-byte 100BASE-X Auxiliary Status Summary register is both readable and writable and is initialized to 0x0000 at device reset.

Table 119: 100BASE-X Auxiliary Status Summary Register Bit Field Descriptions

Field	Name	Description	Access	Default
15	AutoNegComplete (CM)	Auto-Negotiation Complete Returns a 1 if auto-negotiation process has been completed and the contents of AutoNegAdvertise, LinkPartnerAbility, and AutoNegExpansion registers are valid.	R/W	0
14	AutoNegAck Complete (AC)	Auto-Negotiation Acknowledge Complete This read-only field returns a 1 after the acknowledgment exchange portion of the auto-negotiation process has been completed and the arbitrator state machine has exited the complete acknowledge state. It remains this value until the auto-negotiation process is restarted, a Link Fault occurs, auto-negotiation is disabled, or the device is reset.	RO	0
13	AutoNegAclDetect (AD)	Auto-Negotiation Acknowledge Detected This read-only field is set to 1 when the arbitrator state machine exits the acknowledged detect state. It remains high until the auto-negotiation process is restarted, or the device is reset.	RO	0
12	AutoNegAbility Detect (AB)	Auto-Negotiation Ability Detect This read-only field returns a 1 when the auto-negotiation state machine is in the ability detect state. It enters this state a specified time period after the auto-negotiation process begins, and exits after the first FLP burst or link pulses are detected from the link partner. This field returns a 0 any time the auto-negotiation state machine is not in the ability detect state.	RO	0
11	Pause (PS)	Pause Pause operation for full-duplex links. The use of this field is independent of the negotiated data rate, medium, or link technology. The setting of this field indicates the availability of additional DTE capability when full-duplex operation is in use. This field is used by the MAC to communicate pause capability to its link partner and has no effect on PHY operation.	R/W	0
10:8	AutoNegHCD (HC)	Auto-Negotiation HCD This field contains three read-only bits that report the Highest Common Denominator (HCD) result of the auto-negotiation process. Immediately upon entering the link pass state after each reset or restart auto-negotiation, only one of these three bits is set to 1. The link pass state is identified by a 1 in. The HCD bits are reset to 0 every time auto-negotiation is restarted or the device is reset. For their intended application, these bits uniquely identify the HCD only after the first link pass after reset or restart of auto-negotiation. On later Link Fault and subsequent re-negotiations, if the ability of the link partner is different, more than 1 of the above bits can be active.	RO	0

Table 119: 100BASE-X Auxiliary Status Summary Register Bit Field Descriptions (Cont.)

Field	Name	Description	Access	Default
7	AutoNegParallel DetFlt (PF)	Auto-Negotiation Parallel Detection Fault This read-only field gets latched high when a parallel detection fault occurs in the auto-negotiation state machine. For further details, refer to the IEEE standard. This field is cleared to 0 after the register is read, or when the chip is reset.	RO	0
6	LPRemoteFault (RM)	Link Partner Remote Fault This field returns a value of 1 when the link partner signals that a remote fault has occurred. The device simply copies the value to this register and does not act upon it.	R/W	0
5	LPPageRcv (PR)	Link Partner Page Received This field is latched high when a new link code word is received from the link partner, checked, and acknowledged. It remains high until the register is read, or until the chip is reset.	R/W	0
4	LPAutoNegAble (LP)	Link Partner Auto-Negotiation Able This field returns a 1 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 field returns a value of 0.	R/W	0
3	SpeedIndication (SI)	Speed Indication This read-only field shows the true current operation speed of the device. A 1 indicates 100BASE-X operation, and a 0 indicates 10BASE-T. While the auto-negotiation exchange is performed, the device is always operating at 10BASE-T speed.	RO	0
2	LinkStatus (LS)	Link Status The device returns a 1 on this field when the link state machine is in link pass state, indicating that a valid link has been established. Otherwise, it returns 0. When a link failure occurs after the link pass state has been entered, this field is etched at 0 and remains so until the field is read. After the field is read, it becomes 1 when the link pass state is entered again.	R/W	0
1	AutoNegEn (AE)	Auto-Negotiation Enable Auto-negotiation can be disabled by one of two methods: hardware or software control. If the Auto-Negotiation input from the MAC is driven to a logic 0, auto-negotiation is disabled by hardware control. If this field is cleared to 0, auto-negotiation is disabled by software control. When read, this field returns the most recent value, or 1 if auto-negotiation has not been disabled since the last chip reset.	R/W	0
0	JabberDetect (JD)	Jabber Detect 10BASE-T operation only. The device returns a 1 on this field if a jabber condition has been detected. After this field is read once, or if the chip is reset, it reverts to 0.	R/W	0

INTERRUPT REGISTER (INTERRUPT, OFFSET 0X1A)

The two-byte Interrupt register is both readable and writable and is initialized to 0x0F0x at device reset.

Table 120: Interrupt Register Bit Field Descriptions

Field	*Name*	*Description*	*Access*	*Default*
15	Reserved	–	R/W	–
14	InterruptEnable (IE)	Interrupt Enable Setting this field to 1 enables Interrupt Mode.	R/W	–
13:12	Reserved	–	R/W	–
11	Full-Duplex InterruptMask (FD)	Full-Duplex Interrupt Mask When this field is set to 1, changes in Duplex mode do not generate an interrupt.	R/W	–
10	SpeedInterrupt Mask (SI)	Speed Interrupt Mask When this field is set to 1, changes in operating speed do not generate an interrupt.	R/W	–
9	LinkInterrupt Mask (LI)	Link Interrupt Mask When this field is set to 1, changes in Link status do not generate an interrupt.	R/W	–
8	MasterInterrupt Mask (MI)	Master Interrupt Mask When this field is set to 1, no interrupts are generated, regardless of the state of the other mask fields.	R/W	–
7:4	Reserved	–	R/W	–
3	DuplexChange Interrupt (DC)	Duplex Change Interrupt A 1 in this field indicates a change of duplex status since the last register read. A register read clears this field.	R/W	–
2	SpeedChange Interrupt (SP)	Speed Change Interrupt A 1 in this field indicates a change of speed status since the last register read. A register read clears this field.	R/W	–
1	LinkChange Interrupt (LC)	Link Change Interrupt A 1 in this field indicates a change of link status since the last register read. A register read clears this field.	R/W	–
0	InterruptStatus (IS)	Interrupt Status A 1 in this field indicates that the interrupt mask is off and that one or more of the change fields are set. A register read clears this field.	R/W	–

AUXILIARY MODE 2 REGISTER (AUXMODE2, OFFSET 0X1B)

The two-byte Auxiliary Mode 2 register is both readable and writable and is initialized to 0x008A at device reset.

Table 121: Auxiliary Mode 2 Register Bit Field Descriptions

Field	*Name*	*Description*	*Access*	*Default*
15:12	Reserved	–	R/W	–
11	10BTDribbleBit Correct (DC)	10BT Dribble Bit Correct When enabled, the PHY rounds down to the nearest nibble when dribble bits are present on the 10BASE-T input stream.	R/W	–
10	TokenRing (TR)	Token Ring Mode When enabled, the 100BASE-X unlock timer changes to allow long packets.	R/W	–
9	HSTRFIFO (HF)	HSTR FIFO Mode When enabled, the RMII receive FIFO doubles from 7 nibbles to 14 nibbles.	R/W	–
8	Reserved	–	R/W	–
7	BLock10BTEcho (BE)	Block 10BT Echo Mode When enabled, during 10BASE-T half-duplex transmit operation, the MII Transmit Enable input from the MAC does not echo onto the RXDV pin. The MII Transmit Enable input from the MAC echoes onto the CRS pin and the CRS deassertion directly follows the MII Transmit Enable input deassertion.	R/W	–
6	TrafficMeterLED (TM)	Traffic Meter LED Mode When enabled, the Activity LEDs (ACTLED# and FDXLED# if full-duplex LED and interrupt LED modes are not enabled) does not blink based on the internal LED clock (approximately 80 ms on time). Instead, they blink based on the rate of Receive and Transmit activity. Each time a Receive or Transmit operation occurs, the LED turns on for a minimum of 5 ms. During light traffic, the LED blinks at a low rate, while during heavier traffic the LEDs remains on.	R/W	–
5	ActivityLEDForceOn (LF)	Activity LED Force On When asserted, the Activity LEDs (ACTLED# and FDXLED# if full-duplex LED and interrupt LED modes are not enabled) is turned on. This field has a higher priority than the ActivityLEDDisable field in the AuxMode register.	R/W	–
4:3	Reserved	–	R/W	–
2	ActivityLinkLED (AL)	Activity/Link LED Mode When enabled, the ACTLED# output goes active upon acquiring link and pulses during Receive or Transmit activity.	R/W	–

Table 121: Auxiliary Mode 2 Register Bit Field Descriptions (Cont.)

Field	Name	Description	Access	Default
1	QualifiedParallel Detect (QP)	Qualified Parallel Detect Mode This field allows the auto-negotiation/parallel detection process to be qualified with information in the Advertisement Register. • If this field is not set to 1, the local device is enabled to Auto-Negotiate, and the far-end device is a 10BASE-T or 100BASE-X non-Auto-Negotiating legacy type, the local device Auto-Negotiate/Parallel detects the far-end device, regardless of the contents of its AutoNegAdvertise register. • If this field is set to 1, the local device compares the link speed detected to the contents of its Advertisement register. If the particular link speed is enabled in the Advertisement Register, the local device asserts link. If the link speed is disabled in this register, then the local device does not assert link and continues monitoring for a matching capability link speed.	R/W	–
0	Reserved	–	R/W	–

10BASE-T AUXILIARY ERROR AND GENERAL STATUS REGISTER (10BASE-TERRGENSTATUS, OFFSET 0X1C)

The two-byte 10BASE-T Auxiliary Error and General Status register is both readable and writable and is initialized to 0x002x at device reset.

Table 122: 10BASE-T Auxiliary Error and General Status Register Bit Field Descriptions

Field	*Name*	*Description*	*Access*	*Default*
15:14	Reserved	–	R/W	–
13	MDIXStatus (ST)	MDIX Status When this field is set to 1, it indicates that the device has its MDI TD+ and RD+ signals swapped, either due to manually setting MDIXManualSwap field to 1 or through the HP Auto-MDIX function, if it is enabled and the device has detected a MDI cross-over cable.	R/W	–
12	MDIXManualSwap (SW)	MDIX Manual Swap When this field is set to 1, the device forces its MDI TD± and RD± signals to be swapped.	R/W	–
11	HPAutoMDIXDis (HP)	HP Auto-MDIX Disable When auto-negotiation is enabled, this field can be used to disable auto-MDIX via software control. When this field is set to 1, then the BCM4413 disables the HP Auto-MDIX function. When auto-negotiation is disabled, Auto-MDIX is also disabled and this field has no effect.	R/W	–
10	ManCodeErr (MC)	Manchester Code Error When this field is set to 1, it indicates that a Manchester code violation was received. This field is only valid during 10BASE-T operation.	R/W	–
9	EOFError (EF)	End of Frame Error When this field is set to 1, it indicates that the End Of Frame (EOF) sequence was improperly received, or not received at all. This error field is only valid during 10BASE-T operation.	R/W	–
8:4	Reserved	–	R/W	–
3	AutoNegIndicat (AN)	Auto-Negotiation Indication This field indicates whether auto-negotiation has been enabled or disabled on the device. A combination of a 1 in the AutoNegEn field of the MIIControl register and a logic 1 on the Auto-Negotiation input from the MAC is required to enable auto-negotiation. When auto-negotiation is disabled, this field returns a 0. At all other times, it returns a 1.	R/W	–
2	Force100/10Indicat (FI)	Force100/10 Indication This read-only field returns a value of 0 when one of following cases is true: The Auto-Negotiation input from the MAC is low AND the F100 pin is low or the Auto-Negotiation input from the MAC is high AND the AutoNegEn field of the MIIControl register has been written 0 and the ForcedSpeedSel field of the MIIControl register has been written 0. When this field is 0, the speed of the chip is 10BASE-T. In all other cases, either the speed is not forced (auto-negotiation is enabled), or the speed is forced to 100BASE-X.	R/W	–

Table 122: 10BASE-T Auxiliary Error and General Status Register Bit Field Descriptions (Cont.)

Field	Name	Description	Access	Default
1	SpeedIndication (SI)	Speed Indication This read-only field shows the true current operation speed of the device. A 1 indicates 100BASE-X operation, and a 0 indicates 10BASE-T. While the auto-negotiation exchange is performed, the device is always operating at 10BASE-T speed.	R/W	–
0	Full-DuplexIndicat (FD)	Full-Duplex Indication This read-only field returns a 1 when the device is in full-duplex mode. In all other modes, it returns a 0.	R/W	–

AUXILIARY MODE REGISTER (AUXMODE, OFFSET 0X1D)

The two-byte Auxiliary Mode register is both readable and writable and is initialized to 0x000 at device reset.

Table 123: Auxiliary Mode Register Bit Field Descriptions

Field	*Name*	*Description*	*Access*	*Default*
15:5	Reserved	–	R/W	0
4	ActivityLEDDis (AL)	Activity LED Disable When this field is set to 1, it disables the XMTLED# and RCVLED# output pins. When it is cleared to 0, XMTLED# and RCVLED# output pins are enabled.	R/W	0
3	LinkLEDDis (LL)	Link LED Disable When this field is set to 1, it disables the Link LED output pin. When it is cleared to 0, Link LED output is enabled.	R/W	0
2	Reserved	–	R/W	0
1	BlockTXENMode (BT)	Block Transmit Enable Mode When this field is set to 1, Block TXEN mode is enabled. In this mode, short IPGs of 1, 2, 3, or 4 TXC cycles result in the insertion of two IDLEs before the beginning of the next packet's JK symbols.	R/W	0
0	Reserved	–	R/W	0

AUXILIARY MULTIPLE PHY REGISTER (AUXMULTIPHY, OFFSET 0X1E)

The two-byte Auxiliary Multiple PHY register is both readable and writable and is initialized to 0x0000 at device reset.

Table 124: Auxiliary Multiple PHY Register Bit Field Descriptions

Field	Name	Description	Access	Default
15:11	AutoNegHCD (HC)	Auto-Negotiation HCD This field contains five read-only bits that report the Highest Common Denominator (HCD) result of the auto-negotiation process. Immediately upon entering the link pass state after each reset or restart auto-negotiation, only one of these five bits is set to 1. The link pass state is identified by a 1 in the AutoNegAckComplete or the AutoNegComplete field. The HCD bits are reset to 0 every time auto-negotiation is restarted or the device is reset. For their intended application, these bits uniquely identify the HCD only after the first link pass after reset or restart of auto-negotiation. On later Link Fault and subsequent re-negotiations, if the ability of the link partner is different, more than 1 of the above bits can be active.	R/W	0
10:9	Reserved	–	R/W	0
8	RestartAutoNeg (RA)	Restart Auto-Negotiation This self-clearing field allows the auto-negotiation process to be restarted, regardless of the current status of the state machine. For this field to work, auto-negotiation must be enabled. Writing a 1 to this field restarts auto-negotiation. Since the field is self-clearing, it always returns a 0 when read. The operation of this field is identical to the RestartAutoNeg field of the MIIControl register (see "MII Control Register (MIIControl, offset 0x00)" on page 98).	R/W	0
7	AutoNegComplete (CM)	Auto-Negotiation Complete This read-only field returns a 1 after the auto-negotiation process has been completed. It remains 1 until the auto-negotiation process is restarted, a link fault occurs, or the chip is reset. If auto-negotiation is disabled or the process is still in progress, the field returns a 0.	RO	0
6	AutoNegAck Complete (AC)	Auto-Negotiation Acknowledge Complete This read-only field returns a 1 after the acknowledgment exchange portion of the auto-negotiation process has been completed and the arbitrator state machine has exited the complete acknowledge state. It remains this value until the auto-negotiation process is restarted, a Link Fault occurs, auto-negotiation is disabled, or the device is reset.	RO	0
5	AutoNegAclDetect (AD)	Auto-Negotiation Acknowledge Detected This read-only field is set to 1 when the arbitrator state machine exits the acknowledged detect state. It remains high until the auto-negotiation process is restarted, or the device is reset.	RO	0

Table 124: Auxiliary Multiple PHY Register Bit Field Descriptions (Cont.)

Field	Name	Description	Access	Default
4	AutoNegAbility Detect (AB)	Auto-Negotiation Ability Detect This read-only field returns a 1 when the auto-negotiation state machine is in the ability detect state. It enters this state a specified time period after the auto-negotiation process begins, and exits after the first FLP burst or link pulses are detected from the link partner. This field returns a 0 any time the auto-negotiation state machine is not in the ability detect state.	RO	0
3	SuperIsolate (SI)	Super Isolate Writing a 1 to this field places the BCM440X into the Super Isolate mode. Similar to the Isolate mode, all RMII inputs are ignored, and all RMII outputs are tri-stated. Additionally, all link pulses are suppressed. This allows the BCM440X to coexist with another PHY on the same adapter card, with only one being activated at any time.	R/W	0
2	Reserved	–	R/W	0
1	10Base-TSerialMode (SM)	10BASE-T Serial Mode Writing a 1 to bit 1 of the Auxiliary Mode Register enables the 10BASE-T Serial mode. In the normal 10BASE-T mode of operation, as defined by the RMII standard, transmit and receive data packets traverse the TXD1/TXD0 and RXD1/RXD0 busses at a rate of 50 MHz. In the special 10BASE-T Serial mode, data packets traverse to the MAC layer across only TXD0 and RXD0 at a rate of 10 MHz. Serial operation is not available in 100BASE-X mode.	R/W	0
0	Reserved	–	R/W	0

MII SHADOW REGISTERS

The MII shadow registers include the following:

AUXILIARY MODE 4 (SHADOW) REGISTER (AUXMODE4, OFFSET 0X1A)

The two-byte Auxiliary Mode 4 shadow register is both readable and writable and is initialized to 0x0F0x at device reset.

Table 125: Auxiliary Mode 4 Register Bit Field Descriptions

Field	*Name*	*Description*	*Access*	*Default*
15:5	Reserved	–	R/W	–
4	ForceLED (FL)	Force LED [1:0] The 100BASE-X LED., Link Integrity LED, Transmit Activity LED, and Receive Activity LED outputs can be forced to the on state (0) by writing 01 to this field. These LED outputs can be forced to the off state (1) by writing 10 to this field.	R/W	–
3	Reserved	–	R/W	–
2	EnClkLowPower (EC)	Enable Clock During Low Power If this field is set to 1 then the clocks are running in low power mode.	R/W	–
1	ForceLowPower (LP)	Force Low-Power Mode When this field is set to 1, it forces the device to enter low power mode.	R/W	–
0	ForceIDDQ (IQ)	Force IDDQ Mode If this field is set to 1, then the device enters IDDQ mode. When the device is in IDDQ mode, everything is disabled. The device requires a hard reset to return to normal mode.	R/W	–

AUXILIARY STATUS 2 (SHADOW) REGISTER (AUXSTATUS2, OFFSET 0X1B)

The two-byte Auxiliary Status 2 shadow register is both readable and writable and is initialized to 0x008A at device reset.

Table 126: Auxiliary Status 2 Register Bit Field Descriptions

Field	Name	Description	Access	Default
15	MLT3Detected (MD)	MLT3 Detected This field returns a 1 in this field whenever MLT3 signaling is detected.	R/W	–
14:12	CableLength100X (CL)	Cable Length 100X [2:0] This field provides the cable length for each port when a 100TX link is established. The cable length is measured in 20 meter increments, as shown in Table 127.	R/W	–
11:6	ADCPeakAmplitute (PA)	ADC Peak Amplitude [5:0] This field returns the AD converter's six-bit peak amplitude seen during this link.	R/W	–
5	APDEnable (AE)	APD Enable When in normal mode, if this field set to 1, the device enters auto power down mode. If this field is set, the device enters low power mode whenever the link is lost. When energy is detected the device enters full power mode. Otherwise, it wakes up after 2.5s or 5.0s, as determined by the APD Sleep Timer field. When the device wakes up, it sends link pulses and monitors energy. • If the link partner's energy is detected, the device enters full power mode and establishes a link with the link partner. • If no energy is detected from the link partner, it continues to stay in wake-up mode for a duration determined by the APD Wake-up Timer before going to low power mode.	R/W	–
4	ADPSleepTimer (ST)	APD Sleep Timer This field determines how long the device stays in low-power mode before wake-up. If this bit is a 0 then it waits approximately 2.5s before wake-up, else 5s.	R/W	–
3:0	APDWakeupTimer (WT)	APD Wakeup Timer [3:0] This counter determines how long the device stays up in wake-up mode before going to low power mode. This value is specified in 40-ms increments from 0 to 600 ms. A value of 0 forces the device to stay in low power mode indefinitely. In this case the device requires a hard reset to return to normal mode.	R/W	–

Table 127: Cable Length

Cable Length 100x [2:0]	*Cable Length in Meters*
000	< 20
001	20 to <40
010	40 to <60
011	60 to < 80
100	80 to < 100
101	100 to < 120
110	120 to < 140
111	> 140

AUXILIARY STATUS 3 (SHADOW) REGISTER (AUXSTATUS3, OFFSET 0X1C)

The two-byte Auxiliary Status 3 shadow register is both readable and writable and is initialized to 0x002x at device reset.

Table 128: Auxiliary Status 3 Register Bit Field Descriptions

Field	*Name*	*Description*	*Access*	*Default*
15:8	Noise (NS)	Noise [7:0] This field provides the current mean squared error value for noise when a valid link is established.	R/W	–
7:4	Reserved	–	R/W	–
3:0	FIFOConsumption (FC)	FIFO Consumption [3:0] This field indicates the number of nibbles of FIFO currently used.	R/W	–

AUXILIARY MODE 3 (SHADOW) REGISTER (AUXMODE3, OFFSET 0X1D)

The two-byte Auxiliary Mode 3 shadow register is both readable and writable and is initialized to 0x000 at device reset.

Table 129: Auxiliary Mode 3 Register Bit Field Descriptions

Field	Name	Description	Access	Default
15:4	Reserved	–	R/W	0
3:0	–	FIFO Size Select [3:0] This field indicates the current selection of receive FIFO size using bit 3 through 0 as shown in the table below.	R/W	0

Table 130: Current Receive FIFO Size

FIFO Size Select [3:0]	Receive Fifo Size in Use (# of Bits)
0001	16
0010	20
0011	24
0100	28
0101	32
0110	36
0111	40
1000	44
1001	48
1010	52
1011	56
1100	60
1101	64

AUXILIARY STATUS 4 (SHADOW) REGISTER (AUXSTATUS4, OFFSET 0X1E)

The two-byte Auxiliary Status 4 shadow register is both readable and writable and is initialized to 0x0000 at device reset.

Table 131: Auxiliary Status 4 Register Bit Field Descriptions

Field	Name	Description	Access	Default
15:0	PacketLengthCtr (PL)	Packet Length Counter [15:0] This field shows the number bytes in the last packet received. This is valid only when a valid link is established.	R/W	0x0000

DESCRIPTOR PROCESSOR

The Ethernet MAC core contains a single descriptor processor that moves data between the core and memory. The descriptor processor is documented in Section 3: "DMA" on page 31. This section describes core-specific attributes of DMA.

DESCRIPTOR CORE FLAGS

The descriptor processor provides an eight-bit field in each descriptor for use by individual cores for core-specific purposes. The Ethernet MAC core does not use any of the Core Flag descriptor bits.

RECEIVE FRAME DATA HEADER SIZE

The Ethernet MAC core returns an eight-byte Receive Frame Data Header with every frame. The RcvOffset field of the RcvControl register (see "Receive Channel Control Register (RcvControl, offset 0x210)" on page 37) must be set to a value greater than or equal to 8. A descriptor protocol error occurs if the first descriptor fetched for a frame has a buffer size of less than 8 bytes.

DMA-RELATED INTERRUPTS AND ERRORS

The Ethernet MAC core does not have any core-defined DMA-related errors.

ETHERNET MAC INTERFACE

The BCM440X Ethernet Media Access Controller (MAC) provides access to Ethernet physical media through either the internal 10/100 Ethernet transceiver, or an external transceiver through a Media Independent Interface (MII). In addition to basic Ethernet access functions, the MAC provides statistic counters fully compliant with the following Management Information Base (MIB) Statistics standards: RFC 1757, RFC 1643, IEEE 802.3. The MAC features include:

- 802.3u compliant Transmit and Receive engines
- Separate Transmit and Receive FIFOs with programmable watermarks
- Full- and Half-duplex Operation
- Full-duplex frame-based Flow Control compliant with 802.3x
- Internal 64 -entry Content Addressable Memory (CAM) for perfect address filtering
- Automatic CRC checking and generation
- Automatic padding of undersize transmit frames
- MII Management interface allows control of internal and external transceiver functions
- Internal Loopback
- Integrated MIB counters

The Ethernet MAC provides the control and protocol functions necessary for the transmission and reception of 802.3 data streams. During transmission, packet data is removed from the transmit FIFO, framed with preamble and CRC, and forwarded to the transceiver. During reception, the data is received from the transceiver, the frame's destination address and validity is checked, and packet data is placed into the receive FIFO.

ETHERNET PACKET STRUCTURE

The 802.3 Ethernet packet format consists of the following fields:

- **Preamble**—The preamble is a 56-bit field of alternating ones and zeros.
- **Start of Frame Delimiter**—The SFD is the 8-bit field *10101011*, following the preamble.
- **Destination Address**—This field is 48 bits and contains the address of the node to which the frame is addressed. The address can be a unicast address to a specific node, a multicast address to a group of nodes, or a broadcast address to all nodes.
- **Source Address**—This is a 48-bit field and contains the address of the node from which the frame originated.
- **Length/Type**—This is a 16-bit field used as a data length or packet type, depending on the upper layer protocols employed.
- **Data**—The data field is variable in length (46-1500 bytes) and contains the payload information transmitted from one station to another.
- **Frame Check Sequence**—This is a 32-bit field containing the Frame Check Sequence (FCS) for the frame. The FCS is calculated for the entire frame excluding the preamble and SFD using a 32-bit CRC polynomial.

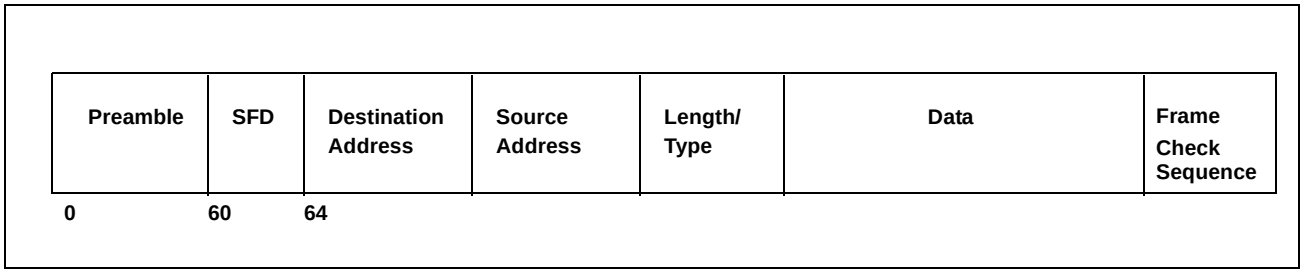

Figure 3: 802.3 Ethernet Frame Format

An Ethernet frame is specified to have a minimum length of 64 bytes and a maximum length of 1518 bytes not including the preamble and SFD.

MAC OPERATION

The Ethernet MAC is organized into three basic sections (see Figure 4):

- Receiver
- Transmitter
- Management information block

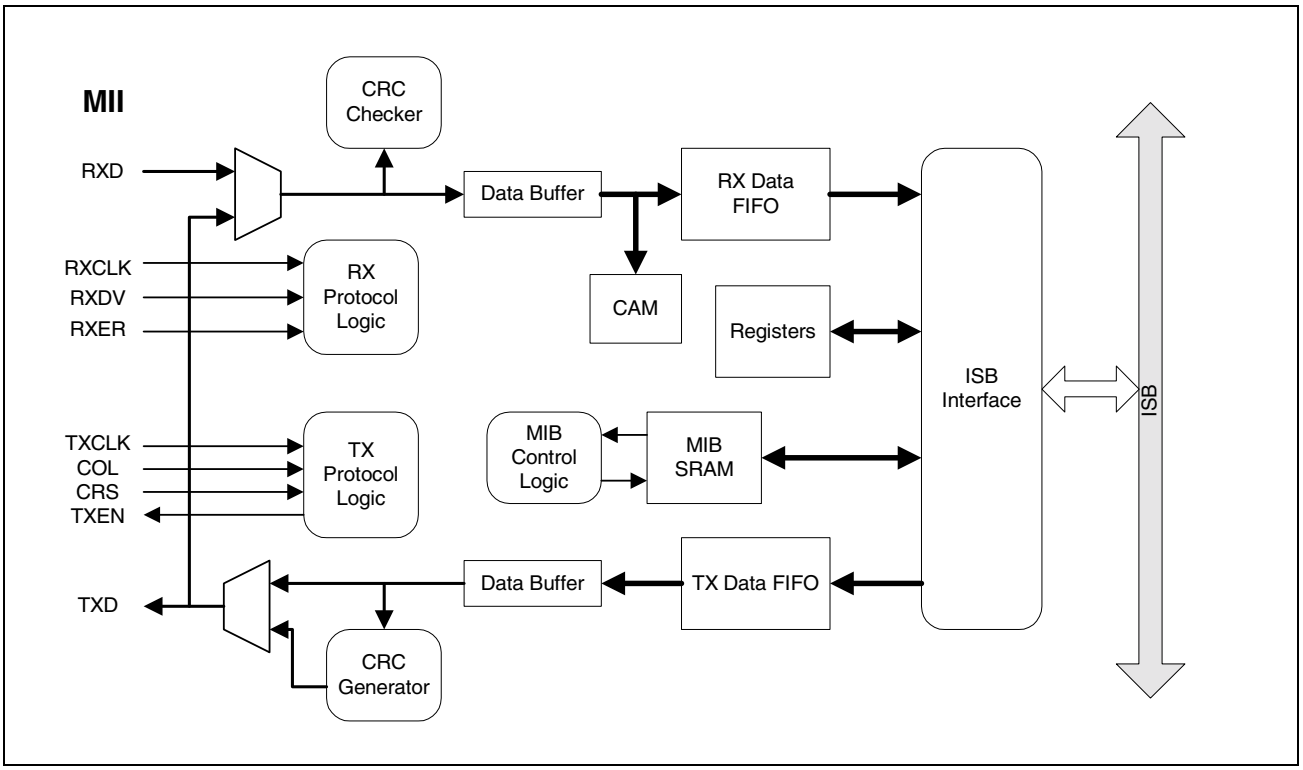

Figure 4: Ethernet MAC

RECEIVER

The receive MAC routes nibble-wide MII data through a buffer where it is assembled into 32-bit-long words. The data is then passed through a 256-byte FIFO, which is connected on the other side to the Internal System Bus (ISB). Ethernet packet data is moved from the FIFO to system memory by the system DMA controller.

The receiver sits idle until RXDV is asserted and the SFD is seen. Once the SFD is seen, the receiver becomes active and begins to process the incoming frame. Processing the frame includes the following functions:

- Address Recognition
- Nibble to Word conversion
- Cyclic Redundancy Check (CRC)
- Status Generation
- Flow Control Identification

Following the End-of-Frame (EOF), the appropriate status is returned to the last receive buffer descriptor in the current frame for use by user software.

The receiver can operate in normal mode or in internal loopback mode. In normal mode, the receiver uses the MII receive interface for accepting packets. When the receiver is placed into loopback mode, the MII transmit interface is connected to the MII receive interface. All MII inputs are ignored (except for the management signals).

The MAC receiver insures the proper sequencing of data during frame reception (loopback or normal). After seeing the SFD, the receiver remains active until RXDV de-asserts. Data is allowed to be placed into the Receive FIFO as long as the receiver is active. Under normal conditions, all frames in which an address match occurs are received into the FIFO. The conditions listed in Table 132 prevent a complete reception of a packet:

Table 132: Receive Packet Discard

Condition	*Description*
Receive FIFO Overflow	The receive FIFO has been completely filled before the device was able to transfer data to memory. The receive FIFO acts as a temporary buffer between the network and the system bus and stores up to 256 bytes of data. A receive overflow condition exists when the FIFO is full and an attempt is made to place more data into the FIFO. When an overflow condition occurs, the current frame is truncated (no more data is placed in the FIFO) and the Overflow bit is asserted in the receive buffer descriptor. No more data is placed into the FIFO until the overrun condition is alleviated. Any incoming frames during this period are lost.
Runt Packet	The received frame is shorter than the minimum required 64 byte length.
Address Mismatch	The packets destination address failed to match one of the stored addresses.

Address Recognition

The receive block contains an internal 64-entry CAM for performing address comparison on unicast and multicast addresses. The broadcast address (all 48 bits are 1) is matched via a hard-wired comparison. The address comparator compares the current frame's destination address (DA) field against the contents of the CAM and the broadcast template to determine whether the incoming frame should be accepted or rejected. The DA can be matched in one of three ways:

- **Unicast Address**—A frame with a unicast address is filtered by comparing the 48-bit destination address against the 48-bit values stored in the internal CAM If a match occurs, the frame is accepted and placed into the receive FIFO. If a match does not occur the frame is rejected.
- **Multicast**—A frame with a multicast address is filtered by applying the 48-bit destination address to a 48-bit wide internal CAM. If the destination address is not stored in the CAM, the frame is rejected. If the destination address is found in the CAM, the frame is accepted and placed into the receive FIFO.
- **Broadcast**—A frame with a broadcast address is always accepted (unless the DisB field in the RcvConfig register is set).

In addition to the address matching mechanisms above, the following modes are also available.

- **Promiscuous Mode**—The receiver receives all frames in promiscuous mode regardless of their destination address. Promiscuous mode is entered by setting the Prom (promiscuous) field in the RcvConfig register.
- **Accept All Multicast**—If the AccMulti field in the RcvConfig register is set, all frames with multicast addresses are accepted.
- **Disable Broadcast**—If the DisB field in the RcvConfig register is set, all frames with broadcast addresses are rejected.

Receive FIFO

The receiver provides a 256-byte FIFO that acts as a buffer and clock boundary between the system bus and the Ethernet MII interface.

Flow Control

The receiver can be configured to receive MAC PAUSE frames to inhibit transmit operation. MAC PAUSE frames and receiver flow control functionality are described in the section entitled Full-Duplex Flow Control.

Receiver Control and Configuration

The receiver is configured and controlled through the RcvConfig register. The fields in this register should only be written to when the Ethernet controller is disabled. Table 133 describes the various modes for which the receiver can be configured.

Table 133: Receiver Modes

Mode	*Description*
Loopback	In loopback mode, the receiver is isolated from the MII interface and transmit data and control is routed to the receiver. No transmit data is passed to the transceiver during loopback.
Receive Disable on transmit (RDT)	In this mode, the receiver is inhibited while transmitting. RDT mode is only valid if the MAC is not in full-duplex mode.
Promiscuous	This mode allows the receiver to accept all frames regardless of the destination address.
Accept All Multicast	This mode allows the receiver to accept all multicast frames.
Reject All Broadcast	This mode forces the receiver to reject all broadcast frames.
Enflow	This mode allows the receiver to process all MAC PAUSE frames. This setting is only valid in full-duplex mode.

TRANSMITTER

The Ethernet transmit MAC supports the following functions:

- Preamble and SFD generation
- CRC generation and append
- Auto padding undersize frames
- Carrier deference
- Interframe spacing
- Collision detection and enforcement
- Collision backoff and retransmission
- Full-duplex and flow control support
- Internal Loopback support
- Error detection and status support
- Transmit FIFO watermark and transmit max length

The transmit MAC contains a 256-byte FIFO, which is connected to the Internal System Bus (ISB) on one side and the MII interface on the other. An Ethernet packet is moved from the system memory into the FIFO by the system DMA controller. The data is then moved from the transmit FIFO through a data buffer to the MII interface by the transmit state machine. The transmit state machine waits to reach the FIFO watermark or the end of a current packet before starting to transmit onto the MII interface. The transmitter also insures that the correct Inter Packet Gap (IPG) duration has expired before allowing the transmission.

The first data transmitted over the 4-bit-wide MII data bus are the Preamble and Start of Frame Delimiter (SFD). Then the actual data is moved out of the 32-bit-wide FIFO through a data buffer onto the MII nibble interface. If the packet is less then 64 bytes long, pad bytes (all zeros) are automatically appended to the frame. The transmitter automatically appends a CRC if the packet is less than 64 bytes long. If the packet is 64 bytes or longer the transmitter appends a CRC only if the CRC_APPEND field in the XmtStatus register in the DMA controller is set (see "Transmit Channel Status Register (XmtStatus, offset 0x20C)" on page 35). The only status returned to the DMA controller is whether the packet transmission encountered a FIFO underrun condition. Otherwise all other status information must be gathered by monitoring the statistic counter registers that are described in detail in "Management Information Block (MIB) Statistic Counters" on page 94.

Preamble and SFD Generation

The transmitter initially sends a 56-bit preamble pattern (alternating ones and zeros) followed by an 8-bit SFD (Start of Frame Delimiter = 10101011) before packet data is sent out onto the MII interface.

CRC Generation

The transmitter automatically generates a CRC derived from the packet data. It appends the CRC to the end of each packet being transmitted if the packet is less than 64bytes long or if it is longer than or equal to 64 bytes and the CRC_APPEND field is set in the XmtStatus descriptor (see "Transmit Channel Status Register (XmtStatus, offset 0x20C)" on page 35) in the DMA controller.

Auto Padding

The transmitter automatically appends a series of octets with an all zero value to the data portion of a MAC frame which is less than 64 bytes long. This function insures that the data portion of a MAC frame is at least 46 bytes long to meet the IEEE 802.3 minimum packet length.

Note: The 4-byte data field is derived by taking the minimum packet size of 64 bytes and subtracting the Destination Address field of six bytes, the Source Address field of six bytes, the Length field of two bytes, and the CRC field of four bytes.

Carrier Deference and Interframe Spacing

Even when it has nothing to transmit the transmit MAC monitors the Carrier Sense input to identify traffic on the network. If the network is busy the transmit MAC defers transmission. After carrier sense is deasserted the transmit MAC starts the interframe timer and continues to defer transmission until the timer times out. The timer is set for 96 bit times (9.6 µs for 10 Mbps, 960 ns for 100 Mbps). If carrier sense becomes activated during the first 2/3 (64-bit times) of the interframe time then the timer is reset. If carrier sense becomes activated during the last 1/3 (32-bit times) of the interframe time then the timer is not reset. If there is a packet to transmit at the end of the interframe spacing the transmit MAC sends it immediately.

Collision Detection and Enforcement

After the deferral described above the transmit MAC can still face contention on the network through a collision. A collision occurs when two nodes begin their transmissions at about the same time. Because of delays in the network the two nodes would not be able to see the others carrier sense until they have already begun their transmissions. In order to handle collisions a value called a slot time is used. A slot time is derived from the sum of the physical layer round-trip propagation time and the maximum jam time. This value has been determined to be 512 bit times (51.2 µs for 10 Mbps, 512 ns for 100 Mbps). Collisions are detected by monitoring the collision detect signal provided by the physical layer over the MII interface. When a collision is detected the MAC sends out a JAM pattern. This collision enforcement or jam guarantees that the duration of the collision is sufficient to ensure its detection by all transmitting stations on the network. The content of the JAM is unspecified but is not equal to the CRC of the fragmented packet. Late collisions are monitored starting at the first bit of preamble.

Collision Backoff and Retransmission

When a transmission attempt has terminated due to a collision, it is retried by the transmit MAC until either it is successful or a maximum number of attempts (16) have been made and all have terminated due to collisions. The scheduling of the retransmissions are controlled by what is called a *truncated binary exponential backoff*. The transmitter delays transmission an integer number of slot times (512 bit times) after the nth retransmission based on a uniformly distributed random integer r in the range:

$$0 \le r < 2^k$$

where

$$k = min(n, 10)$$

If all 16 attempts for transmission fail, the transmitter aborts the transmission and flushes the buffers for that frame from the descriptor chain and it increments a status counter. See "Management Information Block (MIB) Statistic Counters" on page 94 for more details.

Full/Half-Duplex Operation

The transmitter supports both half and full-duplex operation. Half-duplex uses the standard CSMA/CD protocol. Full-duplex operation allows packets to be transmitted and received at the same time. Collisions are ignored in full-duplex mode. Also, if the PHY sets carrier sense during a transmit then it is also ignored.

Flow Control

The transmitter supports flow control for full-duplex operation. The transmitter can be configured to pause its own transmissions if signalled to do so by the receiver. For more information see "Full-Duplex Flow Control" on page 143.

Transmit FIFO

The transmitter supports a 256-byte FIFO that acts as a buffer and clock boundary between the system bus and the Ethernet MII interface. The transmitter stores data from the system bus and holds off transmission until the full packet has been loaded (indicated by an end of frame) or when the programmable transmit watermark (programmed in the XmtWatermark register) has been reached. More details on programming the XmtWatermark register can be found in Section 5: "Ethernet 10/100 Core". The programmable threshold can cover the entire 256 bytes of the FIFO.

Loopback Support

The transmitter supports internal loopback. The user enables loopback mode by setting the Lpbk bit in the RcvConfig (receive configuration) register. While in loopback mode the transmitter ignores the incoming carrier sense (CRS) and collision (COL) signals and disables the outgoing transmit enable (TXEN) and transmit error (TXER) signals. Internally the transmit enable and error signals are routed to the MAC receiver.

Error Detection and Support

The transmitter detects the following errors: underrun, collision 16, and late collision.

The underrun condition occurs when the transmit FIFO becomes empty. This occurs when the system DMA does not provide data in a timely fashion to meet the demand of the Ethernet. The XmtWatermark register can be used to prevent the likelihood of underruns. The higher the Watermark the more time that is given to the system to provide data to the FIFO. The underrun condition is the only status provided on a packet by packet basis in the DMA Transmit Buffer Descriptor.

The collision 16 condition occurs if 16 attempts for transmission fail consecutively due to collisions. The transmitter aborts the transmission and flushes the buffers for that frame from the descriptor chain and it increments a status counter. See "Management Information Block (MIB) Statistic Counters" on page 94 for more details.

A late collision is defined as a collision that occurs outside the first slot time of a packet. A slot time is derived from the sum of the physical layer round-trip propagation time and the maximum jam time. This value has been determined to be 512 bit times (51.2 μs for 10 Mbps, 512 ns for 100 Mbps). A late collision is measured from the first bit of preamble and a counter is incremented for each occurrence. See "Management Information Block (MIB) Statistic Counters" on page 94 for more details.

Watermark and Max Length Support

The **XMTWATERMARK** register represents the number of 32-bit words that are stored in the transmit FIFO before it begins transmitting a frame. The smaller the number programmed in the register the less time the transmitter waits before beginning a transmission. It is very important to note that it is possible to put the transmitter in a locked state if the DMA Maximum Burst Length register for the DMA controller and the XmtWatermark register are not programmed correctly. The watermark must be small enough so that a maximum number of bursts into the FIFO before filling it exceeds the programmed watermark. An equation to use to avoid a failing condition is shown below:

XMIT_WATERMARK <= 63 - (63% DMA_MAX_BURST_LENGTH)

This insures that no matter what size the watermark is set to a discrete number of bursts exceeds the watermark without exceeding the 64 word size of the FIFO.

The XmtMaxLength register represents the number of bytes that constitutes a maximum size Ethernet frame. The default value is 1518 bytes. The purpose for this value is to determine when the status counter for a oversize packet is incremented. See the Management Information Statistic Counters section for more details. Also note that the maximum size frame supported by the transmitter is 2048 bytes. Sending a packet of a size greater than 2048 bytes causes unknown behavior in the transmitter.

CONTENT ADDRESSABLE MEMORY (CAM)

The internal CAM is used to perform address comparison for unicast and multicast addressed packets. The CAM contains 64 identical entries, each of which is simultaneously compared against the incoming packet's destination address. A match on any entry allows the packet to be accepted.

A 48-bit match mask is also supported by the CAM. If a match mask bit is set, then the CAM uses the corresponding address bit in the comparison. If the bit is not set, the address bit is not used in the comparison. This mask applies globally to all CAM entries. Upon reset, all mask bits are set to 1.

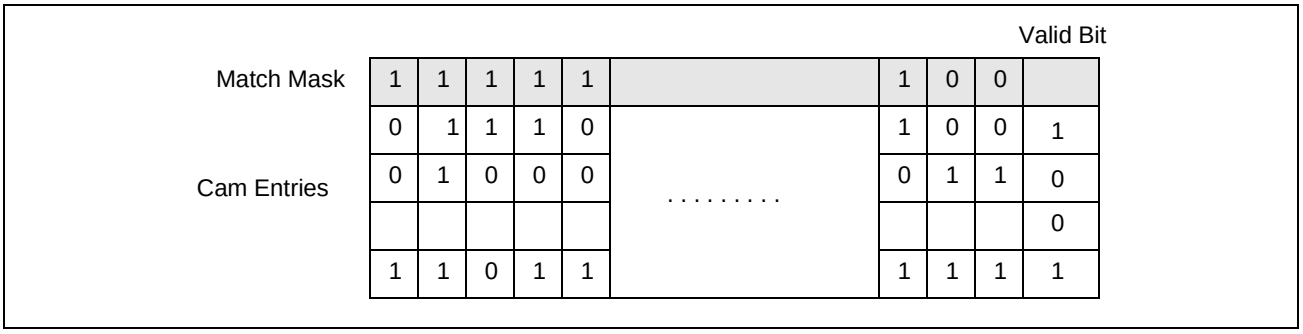

Figure 5: CAM Description

Programming the CAM

Programming the CAM involves the use of three registers. The CAMDataL register contains the lower 32 bits of the destination address to be loaded into the CAM. The CAMDataH register contains the upper 16 bits of the destination address. The CAMControl register contains numerous fields for controlling the type of CAM access and entry location in the CAM. The CAM can only be programmed when it is disabled. To disable the CAM, the CAMEnable field in the CAMControl register must be cleared to 0.

- **Writing** - A write cycle stores the data in the CAMDataL (see "CAM Data Low Register (CAMDataL, offset 0x420)" on page 88) and CAMDataH (see "CAM Data High Register (CAMDataH, offset 0x424)" on page 89) registers in the CAM at the address specified in the Index field. The CAMDataL and CAMDataH registers must first be loaded with the desired data. The CAM entry index and the CAMWrite field are written to the CAMControl register (see "CAM Control Register (CAMControl, offset 0x428)" on page 89). Writing to this register initiates a CAM cycle. Initiating a CAM cycle causes the CAMBusy field to be set to 1. When this field is reset, the CAM cycle is complete and another CAM cycle can be initiated. Software must wait until the CAMBusy field is cleared to 0 before attempting to write to the CAM data or control registers.
- **Reading** - Software can read the CAM by setting CAMRead field and specifying the CAM entry index to be read. Writing to the CAMControl register causes the CAMBusy bit to be asserted and a CAM read cycle to begin. The CAMBusy bit remains asserted until the read cycle is complete. The data read from the CAM is stored in the CAMDataL and CAMDataH registers. Software must wait until the CAMBusy field is cleared to 0 before attempting to write to the CAMData or CAMControl registers.

To direct a read or write operation to the CAM mask register, the MaskSelect field in the CAMControl register must be set.

MANAGEMENT INFORMATION STATISTIC COUNTERS (MIB)

The Ethernet management statistics block contains a set of 47 counters that support the following specifications:

- IETF RFC 1757 (RMON statistics group)
- IETF RFC 1643 (Ethernet-Like MIB)
- Ethernet MIB (IEEE 802.3u)

The statistics block does not provide a unique hardware counter for each statistic required in the above specifications. Unique hardware counters are supplied for IETF 1757 and IETF 1643. The current count for each register is stored in a small 32-bit wide RAM.

Note: Some counter definitions overlap between specifications and two counters can occupy the same physical address. In other words, they are the same counter contained in two different specifications. The counters are listed by specification in Table 134, Table 135 on page 142, and Table 136 on page 142.

Counter Reset

All counters must be reset to 0 by user software.

Counter Reads

When a counter is read, its count is temporarily frozen while the read is in progress. Any requested increments to the counter are delayed until the read of the counter is complete.

Counter Reset On Read

Normally, the counter value is unaffected when a read is performed. By setting the RO field in the MIBControl register, a read to a counter resets its value to 0.

Table 134: RMON Statistics Group (RFC 1757)

#	Name	Description	Rx/Tx
1	rx_etherStatsDropEvents	The total number of packets that experienced an overrun	RX
2	rx_etherStatsOctets	Octets received from the network, including those in bad packets (excludes preamble and SFD bits, includes FCS octets)	RX
3	rx_etherStatsPackets	Good or bad packets received	RX
4	rx_etherStatsBroadcastPkts	Broadcast packets received with a good CRC (excludes multicast packets)	RX
5	rx_etherStatsMuticastPkts	Multicast packets received with a good CRC	RX
6	rx_etherStatsCRCAlignErrors	Packets with a bad CRC or odd number of nibbles	RX
7	rx_etherStatsUndersizePkts	Packets with length < 64 with a good CRC	RX
8	rx_etherStatsOversizePkts	Packets with length > RX_LENGTH and a good CRC	RX
9	rx_etherStatsFragments	Packets with length < 64 and a bad CRC	RX
10	rx_etherStatsJabbers	Packets with length > 1536 and a bad CRC	RX
11	rx_etherStatsPkts64Octets	Good or bad packets 64 bytes in length (excluding preamble)	RX
12	rx_etherStatsPkts65to127Octets	Good or bad packets between 65 and 127 bytes, inclusive, in length (excluding preamble)	RX
13	rx_etherStatsPkts128to255Octets	Good or bad packets between 128 and 255 bytes, inclusive, in length (excluding preamble)	RX
14	rx_etherStatsPkts256to511Octets	Good or bad packets between 256 and 511 bytes, inclusive, in length (excluding preamble)	RX
15	rx_etherStatsPkts512to1023Octets	Good or bad packets between 512 and 1023 bytes, inclusive, in length (excluding preamble)	RX
16	rx_etherStatsPkts1024to1518Octets	Good or bad packets between 1024 and RX_LENGTH bytes, inclusive, in length (excluding preamble)	RX
17	tx_etherStatsOctets	Octets transmitted to the network (excludes preamble and SFD bits, includes FCS octets)	TX
18	tx_etherStatsPackets	All packets transmitted, good or bad	TX
19	tx_etherStatsBroadcastPkts	Good broadcast packets, not including multicast packets	TX
20	tx_etherStatsMulticastPkts	Good multicast packets	TX
--	tx_etherStatsUndersizePkts	Not supported because the transmitter does not transmit undersize packets.	TX
21	tx_etherStatsOversizePkts	Packets with length > TX_MAX_LEN and a good CRC	TX
22	tx_etherStatsFragments	Packets with length < 64 and a bad CRC	TX
23	tx_etherStatsJabbers	Packets with length > TX_MAX_LEN and a bad CRC	TX
24	tx_etherStatsCollisions	The total number of collisions seen by this transmitter	TX
25	tx_etherStatsPkts64Octets	Good or bad packets 64 bytes in length (excluding preamble)	TX
26	tx_etherStatsPkts65to127Octets	Good or bad packets between 65 and 127 bytes, inclusive, in length (excluding preamble)	TX
27	tx_etherStatsPkts128to255Octets	Good or bad packets between 128 and 255 bytes, inclusive, in length (excluding preamble)	TX
28	tx_etherStatsPkts256to511Octets	Good or bad packets between 256 and 511 bytes, inclusive, in length (excluding preamble)	TX

Table 134: RMON Statistics Group (RFC 1757) (Cont.)

#	Name	Description	Rx/Tx
29	tx_etherStatsPkts512to1023Octets	Good or bad packets between 512 and 1023 bytes, inclusive, in length (excluding preamble)	TX
30	tx_etherStatsPkts1024to1518Octets	Good or bad packets between 1024 and TX_MAX_LEN bytes, inclusive, in length (excluding preamble)	TX

Table 135: Ethernet Like MIB (RFC 1643)

#	Name	Description	Rx/Tx
31	dot3StatsAlignmentErrors	Packet received with an odd number of nibbles and a bad CRC	RX
32	dot3StatsFCSErrors	Packets with receive CRC errors	RX
33	dot3StatsSingleCollisionFrames	Packets transmitted successfully with only one collision	TX
34	dot3StatsMultCollisionFrames	Packets transmitted with more than one collision	TX
--	dot3StatsSQE_TestErrors	Not supported	TX
35	dot3DeferredTransmissions	Packet whose transmission is delayed due to a busy network	TX
36	dot3StatsLateCollisions	Number of times a collision occurred after 512 bit times	TX
37	dot3StatsExcessiveCollissions	Packets that attempted more than 15 retries and failed	TX
38	dot3StatsInternalMACTransmitErrors	Packets with a transmit FIFO underflow	TX
39	dot3StatsCarrierSenseErrors	Number of times CRS is lost or never asserted when transmitting a frame	TX
use 8	dot3StatsFrameTooLong	Packets with length > RX_LENGTH	RX
use 1	dot3StatsInternalMACReceiveErrors	Packets with receive FIFO overflow	RX

Table 136: Ethernet MIB (IEEE 802.3u) Selected Counters

#	Name	Description	Rx/Tx
40	aFramesTransmittedOK	The total number of frames that are successfully transmitted	TX
use 33	aSingleCollisionFrames	Packets transmitted successfully with one collision	TX
use 34	aMultipleCollisionFrames	Packets transmitted successfully with more than one collision	TX
41	aFramesReceivedOK	The total number of frames that are successfully received	RX
use 32	aFrameCheckSequenceErrors	The total number of frames that are an integral number of bytes and do not pass the FCS check	RX
use 31	aAlignmentErrors	The total number of frames that are not an integral number of bytes and do not pass the FCS check	RX
42	aOctetsTransmittedOK	The total number of octets transmitted in good packets	TX
use 35	aFramesWithDeferredXmissions	The count of frames whose transmission was delayed on its first attempt because the network was busy	TX
use 36	aLateCollisions	Number of times a collision occurred after 512 bit times	TX
use 37	aFramesAbortedDueToXSColls	Packets that attempted more than 15 retries and failed	TX
use 38	aFramesLostDueToIntMACTxE	Packets with a transmit FIFO underflow	TX
use 39	aCarrierSenseErrors	Number of times CRS is lost or never asserted when transmitting a frame.	TX

Table 136: Ethernet MIB (IEEE 802.3u) Selected Counters (Cont.)

#	Name	Description	Rx/Tx
43	aOctetsReceivedOK	The total number of octets received in good packets	RX
use 1	aFrameLostDueToIntMACRcErr	Packets with receive FIFO overflow	RX
use 20	aMulticastFramesXmittedOK	Good multicast packets	TX
use 19	aBroadcastFramesXmittedOK	Good broadcast packets, not including multicast packets	TX
--	aFramesWithExcessiveDeferral	Not supported	TX
use 5	aMulticastFramesReceivedOK	Multicast packets received with a good CRC	RX
use 4	aBroadcastFramesReceivedOK	Broadcast packets received with a good CRC (excludes multicast packets)	RX
44	aSymbolErrorDuringCarrier	RXER occurs while CRS is asserted	RX
45	aPauseMACControlFramesRcvd	MAC control frames with Pause opcode	RX
46	aMACControlFramesRcvd	MAC control frames without Pause opcode	RX
47	aPauseMACControlFramesX	MAC control frames with Pause opcode transmitted	TX

FULL-DUPLEX FLOW CONTROL

The MAC PAUSE frame defined in 802.3x (clause 31) was established for providing a means of framed-based flow control between two nodes operating in full-duplex mode. The MAC PAUSE frame is a type of MAC CONTROL frame which specifies a pause interval for which the receiving node is required to inhibit it's transmissions. The Ethernet MAC can be configured to automatically receive and transmit MAC PAUSE frames during full-duplex operation.

The MAC PAUSE frame is identified via the packet structure shown in Table 137.

Table 137: MAC Control Frame Format

Length	Description (PAUSE Specific)
7 bytes	Preamble
1 byte	SFD
6 bytes	Destination Address (01-80-C2-00-00-01)
6 bytes	Source Address
2 bytes	Length/Type (88-08)
2 bytes	Opcode (00-01)
2 bytes	Parameters (PAUSE_TIME)
42 bytes	Reserved
4 bytes	Frames Check Sequence

The MAC can be configured to support flow control in the following ways:

- **Flow control disabled**—any received PAUSE frames are ignored.
- **Asymmetric flow control**— reception of PAUSE frames is enabled.

The reception of MAC PAUSE frames is controlled with configuration registers. The default mode of the MAC is to disable the reception and transmission of PAUSE frames. To enable the receiver to accept PAUSE frames, the EnFlow field in the RcvConfig register (see "Receiver Configuration Register (RcvConfig, offset 0x400)" on page 85) must be set. The EnFlow field is only valid when the FD bit is set in the TX_CONFIG register.

Pause Frame Reception and Transmitter Pause

As a frame is received, the destination address, length/type, and opcode fields are compared against the values specified in the above table. If the fields match and the frame is good (valid CRC, 64 bytes in length), a valid PAUSE frame is recognized and the two-byte PAUSE_TIME value is loaded into the transmit pause timer.

Once a non-zero value has been loaded into the transmit pause timer, transmission of packets by the MAC is inhibited until the timer expires. If a PAUSE frame is received while another packet is being transmitted, then the current transmission is completed before the pause interval takes effect. During the pause interval, packet data for subsequent transmissions is held in the TX FIFO until the interval expires.

The pause timer can expire in two ways: by allowing it to auto-decrement until it reaches zero, or via the reception of a subsequent MAC PAUSE frame with a PAUSE_TIME value of 0. The 16-bit PAUSE_TIME field can have a minimum value of 0 and a maximum value of 65535. Each unit is equal to one slot time or 512 bit times. The maximum interval that the MAC transmitter can be paused is 335.54 ms at 100 Mbps or 3355.4 ms at 10 Mbps.

PAUSE frames are absorbed by the receiver and are not placed in the receive data FIFO. If a PAUSE frame is received while in half-duplex mode, it is ignored by the flow control logic and is placed into the receive data FIFO.

INTERRUPTS

The MAC provides support for the three interrupts shown in Table 138.

Table 138: MAC Interrupts

Source	*Description*
MII	After the completion of a MII management write or read operation, the MII interrupt is generated.
MIB	When any MIB statistics counter reaches 50% of its maximum value, a MIB interrupt is generated
Flow Control	When the number of available receive buffer descriptors falls below the specified threshold, a flow control interrupt is generated.

Each interrupt can be turned off by setting the corresponding mask bit in the EnetIntMask register (see "Ethernet Interrupt Mask Register (EnetIntMask, offset 0x418)" on page 87) to 0. The power-up value of all mask bits is 0. Refer to the section on register descriptions for a detailed explanation of the EnetIntMask register and the EnetIntMask register.

ETHERNET CORE INITIALIZATION

To prevent error conditions on startup, software should initialize the Ethernet MAC core by performing the following steps:

1. Enable clocks and reset the core using the sequence of write to the SBTMSTATELOW register (see "System Backplane Target State Low Register (SBTMSTATELOW, offset 0xF98)" on page 17) described in "Reset" on page 203.
2. Initialize the descriptor processor as described in "Descriptor Processor Channel Initialization" on page 45.
3. Modify registers in the Ethernet MAC space (see "Ethernet MAC Registers" on page 58, offsets 0x400-0x5FF), if desired.
4. Set the EmacEnable field of the EnetControl register (see "Ethernet Control Register (EnetControl, offset 0x42C)" on page 90) to 1, enabling the MAC.

Before making further modifications to Ethernet MAC space registers, the MAC should be disabled by writing a 1 to the EmacDisable field of the EnetControl register. Once registers have been modified, the EmacEnable field of the EnetControl register should be set to 1.

The MAC can also be internally reset by setting the EMACSoftReset field of the EnetControl register to 1. The software driver should poll the EMACSoftReset field waiting for it to go low before accessing the block. Accessing the MAC before the EMACSoftReset field has been cleared by internal logic can cause unpredictable behavior.

Access to PHY registers using the MII management interface can be done regardless of the Ethernet MAC enable state.

NETWORK WAKEUP FRAME DETECTION

The Ethernet MAC core signals a power management event upon the reception of an Ethernet frame when these conditions are satisfied:

- Power management event signaling is enabled by the PMEEnable field of the SBTMSTATELOW register (see "System Backplane Target State Low Register (SBTMSTATELOW, offset 0xF98)" on page 17).
- Received frame has a valid FCS field.
- Destination Ethernet address of the received field successfully passes through the address filtering stage of the Ethernet Frame Processor.
- Received frame matches one of the patterns stored in the PME Frame Match Table or the PatMatchEn field of the DevControl register (see"Device Control Register (DevControl, offset 0x00)" on page 62) is clear.
- The received frame contains an AMD® Magic Packet sequence (BCM4401 B0 and later only)

WAKEUP FRAME PATTERNS

The BCM440X contains four (or Interesting Packets) 128-byte patterns that are compared against the start of a received frame. Each pattern has an associated 128-bit match enable vector, so that an individual byte in the pattern matches a packet if its match enable bit is cleared to 0, or if the match enable bit is set to 1 *and* the received frame's data byte equals the pattern's data byte. Each pattern has an associated minimum length field in the WakeupLength register (see "Wakeup Length Register (WakeupLength, offset 0x10)" on page 66). An incoming frame matches a pattern only if it is longer than the pattern's length and matches the first length bytes of the pattern.

The BCM440X stores the network wakeup frame patterns and pattern match enable vectors in the Ethernet filter table. The patterns are stored into the chip using the ENetFtAddr (see "Ethernet Filter Table Address Register (ENetFtAddr, offset 0x90)" on page 76) and ENetFtData (see "Ethernet Filter Table Data Register (ENetFtData, offset 0x94)" on page 77) registers. All four frame patterns must be initialized before the network wakeup frame filtering is enabled (by setting the PatMatchEn field in DevControl register, see "Device Control Register (DevControl, offset 0x00)" on page 62). Unused patterns can be disabled by setting the associated DisableN field in the WakeupLength register. When network wakeup frame filtering is disabled by clearing the PatMatchEn field in the DevControl register, any valid frame is passed through as if it matched one or more patterns.

For each of the four wakeup patterns, the corresponding wakeup pattern length field in the WakeupLength register must be set to be equal to one less than the number of bytes that are significant in the wakeup pattern. This is a value from 0 to 127. The following code fragment depicts the algorithm used to determine if a valid received frame asserts a power management event:

```
doPME =!devcontrol.PatMatchEn;
if (!doPME) {
    for (i=0;i<4;i++) {
        if (!WakeupLength.disable(i) &&
        Length(IncomingPacket)>WakeupLength.pattern(i)){
            doPME = TRUE;
            for (j=0;j<=WakeupLength.pattern(i) && doPME;j++)
                doPME &= (PatternMask(i,j) == 0) ||
                    (Pattern(i,j)==IncomingPacket(j));
            if (doPME) break;
        }
}
if (doPME && PMCSR.PMEEn)
    AssertSignal(PME);
```

AMD® MAGIC PACKET WAKEUP

The BCM4401 B0 and later can be configured to support wakeup from an Ethernet frame that contains an AMD Magic Packet in addition to the Wakeup Frames (or Interesting Packets). To enable Magic Packet wakeup on the BCM4401 B0 or later adapters, the ENetMPAddrUpper and ENetMPAddrLowe registers must be programmed with the local MAC address (see "Ethernet Magic Packet™ Address Lower Register (ENetMPAddrLower, offset 0x88) (BCM4401 B0 only)" on page 75). The Magic Packet functionality must be enabled by setting the MagicPME bit and the PatMatchEn bit of the DevControl register ("Device Control Register (DevControl, offset 0x00)" on page 62).

If the PatMatchEn bit is set and the BCM440X has not also been programmed for Interesting Packet wakeup, the Wakeup Length Register ("Wakeup Length Register (WakeupLength, offset 0x10)" on page 66) must be set so that other packets will not wake the system by programming the value 0x80808080.

Section 6: PCI Core

Modes and Functions

There is no DMA processor in the PCI core. Data movement between the PCI bus and other cores is initiated by external devices or DMA processors in other cores.

Address Spaces

System Backplane

The PCI core defines two system backplane address spaces:

- enumeration space
- a 128 MB PCI access space

The enumeration space includes registers used for GPIOs and interrupts, as well as the standard backplane configuration registers.

The smaller PCI access space maps 128 MB of backplane address space onto the PCI bus. The base address of this region is set by the SBADDRMATCH1 register (see "System Backplane Address Match 1 Register (SBADMATCH1, offset 0xF70)" on page 14) in the core's enumeration space. Two SBtoPCITranslation registers (see "Address Translation Registers" on page 159) each map 64 MB of PCI bus space and can map internal accesses on to PCI memory, I/O, or configuration accesses. PCI memory accesses can be performed as burst; I/O and configuration accesses cannot be bursts.

The PCI core's local configuration space cannot be accessed directly from the backplane. These accesses must be performed externally with PCI configuration cycles.

PCI Bus

Each PCI function contains three address regions that can be accessed from the PCI bus:

- configuration space
- register space
- memory space

Each function's configuration space contains two memory BARs, a power management capability block, and several device-specific registers. The extra registers are used to define PCI windows onto the internal backplane's address space and pass interrupt information between internal cores and the PCI bus.

Each function's BAR0 register (see "Base Address 0 Register (BAR0, offset 0x10)" on page 169) maps an 8K core register region that should be used to access the enumeration space of a single core. Accesses to the lower half of the core register region are translated into system backplane accesses using the PCIBar0Windo**w** register (see "PCI to Bar 0 Window Register (PCIBar0Window, offset 0x80)" on page 176) defined in "Device Specific Configuration Space Registers" on page 175.

Accesses to the space defined by BAR0 are always enabled and are disconnected after a single data transfer.

Each function has a BAR1 memory space intended for PCI accesses to the larger region of the backplane address space. Accesses to the BAR1 memory space are burstable and are translated into system backplane accesses using the PCIBar1Window register (see "PCI to Bar 1 Window Register (PCIBar1Window, offset 0x84)" on page 176) defined in "Device Specific Configuration Space Registers" on page 175.

Byte swapping can be performed on DMA accesses to and from an external device by mapping the BAR1 region onto the SDRAM controller's big-endian space.

Host control registers (see "Enumeration Space and Host Control Registers" on page 150) in the PCI core's enumeration space can not be accessed directly from the PCI bus. These accesses must be performed by mapping one of the BARs onto the enumeration space. Writes to the Expansion ROM must be separated by a minimum of 10 ms.

EXPANSION ROM

The lPCI core supports an expansion ROM. When the enable field of the ExpansionRomBAR configuration space register and the MemorySpace field of the Command register (see "Command Register (Command, offset 0x04)" on page 165) are both set to 1, reads to the address space defined by the ExpansionRomBAR register (see "Expansion ROM Bar Register (ExpansionROMBar, offset 0x30)" on page 171) returns the contents of the external Boot ROM. When the BootROMWriteEn field of the SpromCt register is also set, writes to that address space modify the contents of the Boot ROM.

The size of the Boot ROM address space is determined by the BootROMSize field of the SPROM.

The core supports 1, 2, and 4 byte reads and 1 byte writes.

INTERRUPTS

The PCI core provides interrupt mailboxes that can be used to raise software-defined interrupts from PCI to the system backplane or from the system backplane to PCI. Each PCI function provides two mailbox interrupts in each direction. The system backplane writes to the SBtoPCIMailbox register (see "System Backplane to PCI Mailbox Register (SBtoPCIMailbox, offset 0x28)" on page 155) are reflected in the PCIIntStatus register (see "PCI Interrupt Status Register (PCIIntStatus, offset 0x90)" on page 179) in configuration space. If the corresponding fields are set in the PCIIntMask configuration register (see "PCI Interrupt Mask Register (PCIIntMask, offset 0x94)" on page 180), the PCI_INTA_L signal is asserted. Configuration writes to the PCItoSBMailbox register are reflected in the IntStatus register. If the corresponding fields are set in the IntMask register, the PCI core's backplane interrupt is asserted.

Enumeration Space and Host Control Registers

This section describes the registers in the PCI core's enumeration space. These registers are accessible only from the system backplane. PCI bus accesses can be performed by mapping one of the BARs onto the enumeration space. All of the registers are four bytes wide, but they can be accessed at any size.

The enumeration space and host control registers include the following register sets:

Core Control and Status Registers

The following registers are used for core control and status:

BUILT-IN SELF TEST REGISTER (BISTSTATUS, OFFSET 0X0C)

The four-byte BISTStatus register is readable and contains the value 0x0 at device reset.

Table 139: Built-in Self Test Status Register Bit Field Descriptions

Field	Name	Description	Access	Default
31:4	Reserved	–	–	0
3:0	BISTFail (BF)	Built-In Self Test Fail This field contains the result of the Built-In Self Test for all of the RAMs in the PCI core. If any bit is set to 1, the BIST failed for the corresponding RAM. See Table 46 for the mapping of RAMs to bits within this field. This field is valid only when the BistDone field is set in the SBTMSTATEHIGH register (see "System Backplane Target State High Register (SBTMSTATEHIGH, offset 0xF9C)" on page 18).	–	0

Table 140: RAM-to-Bit Mapping

Bit	RAM
0	Receive DMA FIFO
1	Transmit DMA FIFO

INTERRUPT CONTROL REGISTERS

The following registers are used for interrupt control:

INTERRUPT STATUS REGISTER (INTSTATUS, OFFSET 0X20)

The four-byte Interrupt Status register is both readable and writable and contains the value 0x0 at device reset. Fields in this register indicate that interrupts are pending, but interrupts are not signaled on the system backplane unless the corresponding enable field in the IntMask register is also set.

Table 141: Interrupt Status Register Bit Field Descriptions

FIeld	*Name*	*Description*	*Access*	*Default*
31:12	Reserved	–	R/W	0
11:10	Function1 (F1)	Function 1 Mailbox This field is set to 1 by writing the PCItoSBMailbox register in function 1s configuration space. The bits in this field are cleared by writing ones to them.	R/W	0
9:8	Function0 (F0)	Function 0 Mailbox This field is set to 1 by writing the PCItoSBMailbox register in function 0s configuration space. The bits in this field are cleared by writing ones to them.	R/W	0
7:5	Reserved		R/W	0
4	PCIPME (PM)	PCI_PME_L This read-only field is set to 1 when the PCI_PME_L pin is asserted and is cleared to 0 if PCI_PME_L is not asserted.	R/W	0
3	PCIPerr (PP)	PCI_PERR_L This field is set to 1 when the core observes an assertion of the PCI_PERR_L pin. Writing a 1 to this field clears the field and the pending interrupt.	R/W	0
2	PCISerr (PS)	PCI_SERR_L This field is set to 1 when the core observes an assertion of the PCI_SERR_L pin. Writing a 1 to this field clears the field and the pending interrupt.	R/W	0
1	PCIIntB (PB)	PCI_INTB_L This read-only field is set to 1 when the PCI_INTA_L pin is asserted and is cleared to 0 if PCI_INTA_L is not asserted.	R/W	0
0	PCIIntA (PA)	PCI_INTA_L This read-only field is set to 1 when the PCI_INTA_L pin is asserted and is cleared to 0 if PCI_INTA_L is not asserted.	R/W	0

INTERRUPT MASK REGISTER (INTMASK, OFFSET 0X24)

The four-byte Interrupt Mask register is both readable and writable and contains the value 0x0 at device reset. The fields in this register selectively enable the propagation of interrupt conditions onto the system backplane.

Table 142: Interrupt Mask Register Bit Field Descriptions

FIeld	*Name*	*Description*	*Access*	*Default*
31:12	Reserved	–	R/W	0
11:10	Function1 (F1)	Function 1 Mailbox When set to 1, the bits in this field enable interrupts from the function 1 mailbox	R/W	0
9:8	Function0 (F0)	Function 0 Mailbox When set to 1, the bits in this field enable interrupts from the function 0 mailbox.	R/W	0
7:5	Reserved	–	R/W	0
4	PCIPME (PM)	PCI_PME_L When set to 1, this field enables interrupts caused by assertions of the PCI_PME_L pin.	R/W	0
3	PCIPerr (PP)	PCI_PERR_L When set to 1, this field enables interrupts caused by assertions of the PCI_PERR_L pin.	R/W	0
2	PCISerr (PS)	PCI_SERR_L When set to 1, this field enables interrupts caused by assertions of the PCI_SERR_L pin.	R/W	0
1	PCIIntB (PB)	PCI_INTB_L When set to 1, this field enables interrupts caused by assertions of the PCI_INTB_L pin.	R/W	0
0	PCIIntA (PA)	PCI_INTA_L When set to 1, this field enables interrupts caused by assertions of the PCI_INTA_L pin.	R/W	0

SYSTEM BACKPLANE TO PCI MAILBOX REGISTER (SBTOPCIMAILBOX, OFFSET 0X28)

The four-byte System Backplane to PCI Mailbox register is write-only. This register is used to raise PCI bus interrupts. Fields in this register are cleared when they are read from the destination function's configuration space.

Table 143: System Backplane to PCI Mailbox Register Bit Field Descriptions

FIeld	*Name*	*Description*	*Access*	*Default*
31:12	Reserved	–	WO	–
11:10	Function1 (F1)	Function 1 Mailbox Bits in this field are set to 1 to raise PCI bus interrupts through function 1 configuration space.	WO	–
9:8	Function0 (F0)	Function 0 Mailbox Bits in this field are set to 1 to raise PCI bus interrupts through function 0 configuration space.	WO	–
7:0	Reserved	–	WO	–

BROADCAST REGISTERS

The Broadcast registers are used to perform system backplane broadcast requests. See "Updating Buffered Registers" on page 11 for information on using broadcast requests.

The following registers are used for broadcast requests:

- "Broadcast Address Register (BroadcastAddress, offset 0x50)" on page 156
- "Broadcast Data Register (BroadcastData, offset 0x54)" on page 156

BROADCAST ADDRESS REGISTER (BROADCASTADDRESS, OFFSET 0X50)

The four-byte Broadcast Address register is both readable and writable and contains the value 0x0 at device reset. It is used to generate the address of a broadcast request on the system backplane. When the BroadcastData register (see "Broadcast Data Register (BroadcastData, offset 0x54)" on page 156) is written, bits [11:0] of this register are driven as bits [11:0] of the address of the broadcast request.

BROADCAST DATA REGISTER (BROADCASTDATA, OFFSET 0X54)

The four-byte, write-only System Backplane Broadcast Data register contains the value 0x0 at device reset. Writes to this register generate a broadcast request on the system backplane using the data written to this register and the address contained in the BroadcastAddress register (see "Broadcast Address Register (BroadcastAddress, offset 0x50)" on page 156).

GPIO REGISTERS

Bits [7:0] of the GPIOInput, GPIOOutput, and GPIOOutEn registers correspond to GPIO pins [7:0]. Bits [31:8] are reserved. The GPIOControl register contains one bit for all 16 GPIO pins in bits [15:0]. Some of the GPIO pins are not available in all configurations. See the device data sheet for more information.

The following registers are used for GPIO:

- "General Purpose I/O Input Register (GPIOInput, offset 0x60)" on page 158
- "General Purpose I/O Output Register (GPIOOutput, offset 0x64)" on page 158
- "General Purpose I/O Output Enable Register (GPIOOutEn, offset 0x68)" on page 158
- "General Purpose I/O Control Register (GPIOControl, offset 0x6C)" on page 158

For the BCM440 B0 only:

- "General Purpose I/O Input Register (GPIOInput, offset 0xB0) (BCM4401 B0 Only)" on page 158
- "General Purpose I/O Output Register (GPIOOutput, offset 0x0xB4) (BCM4401 B0 Only)" on page 158
- "General Purpose I/O Output Enable Register (GPIOOutEn, offset 0x0xB8) (BCM4401 B0 Only)" on page 158

GENERAL PURPOSE I/O INPUT REGISTER (GPIOINPUT, OFFSET 0X60)

The four-byte General Purpose I/O Input register is read-only and always contains the values that are present on the GPIO pins, including those being driven by the device.

GENERAL PURPOSE I/O OUTPUT REGISTER (GPIOOUTPUT, OFFSET 0X64)

The four-byte General Purpose I/O Output register is both readable and writable and contains the value 0x0 at power-on reset. This register contains an arbitrary value that is placed on the GPIO pins. Each pin is driven only when the corresponding bit in the GPIOOutEn register is set to 1. When read, this register returns the value last written. The GPIO output values resulting from combining this register with the GPIOOutEn register can be read from the GPIOInput register.

GENERAL PURPOSE I/O OUTPUT ENABLE REGISTER (GPIOOUTEN, OFFSET 0X68)

The four-byte General Purpose I/O Output Enable register is both readable and writable and contains the value 0x0 at power-on reset. This register determines which GPIO pins are used as inputs and which are used as outputs. GPIO pin X is driven as an output whenever bit X of the GPIOOutEn register is set to 1 and is received as an input whenever bit X of the GPIOOutEn register is set to 0.

GENERAL PURPOSE I/O CONTROL REGISTER (GPIOCONTROL, OFFSET 0X6C)

The four-byte General Purpose I/O Control register is both readable and writable and contains the value 0x0 at power-on reset. This register is necessary on chips where the GPIO signals are multiplexed with other signals before reaching external signal pins. GPIO pin X is assigned to the PCI core when bit X of this register is set to 0. It is assigned to a chip-dependent value when bit X of this register is set to 1. Bits [15:0] of these registers correspond to GPIO pins [15:0].

GENERAL PURPOSE I/O INPUT REGISTER (GPIOINPUT, OFFSET 0XB0) (BCM4401 B0 ONLY)

The four-byte General Purpose I/O Input register is read-only and always contains the values that are present on the GPIO pins, including those being driven by the device.

GENERAL PURPOSE I/O OUTPUT REGISTER (GPIOOUTPUT, OFFSET 0X0XB4) (BCM4401 B0 ONLY)

The four-byte General Purpose I/O Output register is both readable and writable and contains the value 0x0 at power-on reset. This register contains an arbitrary value that is placed on the GPIO pins. Each pin is driven only when the corresponding bit in the GPIOOutEn register is set to 1. When read, this register returns the value last written. The GPIO output values resulting from combining this register with the GPIOOutEn register can be read from the GPIOInput register.

GENERAL PURPOSE I/O OUTPUT ENABLE REGISTER (GPIOOUTEN, OFFSET 0X0XB8) (BCM4401 B0 ONLY)

The four-byte General Purpose I/O Output Enable register is both readable and writable and contains the value 0x0 at power-on reset. This register determines which GPIO pins are used as inputs and which are used as outputs. GPIO pin X is driven as an output whenever bit X of the GPIOOutEn register is set to 1 and is received as an input whenever bit X of the GPIOOutEn register is set to 0.

ADDRESS TRANSLATION REGISTERS

The following registers are used for address translation:

SYSTEM BACKPLANE TO PCI TRANSLATION 0 REGISTER (SBTOPCITRANSLATION0, OFFSET 0X100)

The four-byte System Backplane to PCI Translation 0 register is both readable and writable and contains the value 0x0 at device reset. The contents of this register are used to generate the upper bits of PCI bus addresses and the PCI access type for backplane accesses to the lower 64 MB of the host mode PCI access space.

Table 144: System Backplane to PCI Translation 0 Register Bit Field Descriptions

Field	Name	Description	Access	Default
31:26	UpperAddress (UA)	Upper Address PCI bus addresses for PCI accesses mapped by this register are formed by concatenating the contents of this field with bits [25:0] of the backplane address being translated.	R/W	0
25:4	Reserved	–	R/W	0
3	WriteBurstEn (WB)	Write Burst Enable When this field is set to 1, PCI memory write accesses mapped by this register can generate PCI burst transfers. When cleared to 0, all PCI writes are performed as single-word transactions. Configuration and I/O writes are never bursted. Setting this field to 1 improves system performance.	R/W	0
2	PrefetchEn (PE)	Prefetch Enable When this field is set to 1, PCI memory accesses mapped by this register can cause additional data to be prefetched. When cleared to 0, prefetches are never generated. Configuration and I/O access are never mapped. Setting this field to 1 improves system performance.	R/W	0
1:0	AccessType (AT)	Access Type The contents of this field determine the type of PCI access that is generated on references mapped by this register. The mapping of this field is defined in Table 145.	R/W	0

Table 145: Access Types

Value	Access Type
0	Read or Write
1	I/O Read or I/O Write
2	Type 0 Configuration Read or Write
3	Type 1 Configuration Read or Write

SYSTEM BACKPLANE TO PCI TRANSLATION 1 REGISTER (SBTOPCITRANSLATION1, OFFSET 0X104)

The four-byte System Backplane to PCI Translation 1 register is both readable and writable and contains the value 0x2 at device reset. The contents of this register are used to generate the upper bits of PCI bus addresses and the PCI access type for backplane accesses to the upper 64 MB of the host mode PCI access space.

Table 146: System Backplane to PCI Translation 1 Register Bit Field Descriptions

Field	Name	Description	Access	Default
31:26	UpperAddress (UA)	Upper Address PCI bus addresses for PCI accesses mapped by this register are formed by concatenating the contents of this field with bits [25:0] of the backplane address being translated.	R/W	–
25:4	Reserved	–	R/W	–
3	WriteBurstEn (WB)	Write Burst Enable When this field is set to 1, PCI memory write accesses mapped by this register can generate PCI burst transfers. When cleared to 0, all PCI writes are performed as single-word transactions. Configuration and I/O writes are never bursted. Setting this field to 1 improves system performance.	R/W	–
2	PrefetchEn (PE)	Prefetch Enable When this field is set to 1, PCI memory accesses mapped by this register can cause additional data to be prefetched. When cleared to 0, prefetches are never generated. Configuration and I/O access are never mapped. Setting this field to 1 improves system performance.	R/W	–
1:0	AccessType (AT)	Access Type The contents of this field determine the type of PCI access that is generated on references mapped by this register. The mapping of this field is defined in Table 145 on page 160.	R/W	–

SPROM Shadow Area (SPROMShadow, Offset 0x800 to 0x847)

Addresses from 0x800 to 0x837 (0x847 for the BCM4401 B0 only) are used to shadow the contents of the first 56 bytes (72 bytes for the BCM4401 B0 only) of the SPROM. There are four (two-byte) words of common area followed by eight (two-byte) words for each PCI function on the BCM440X. This region accesses the PCI configuration state that wold normally be read from an external SPROM. This space allows a device, where no SPROM is present, to initialize its configuration space. See Table 163: "SPROM Layout (BCM4401 B0 and later)," on page 206 for a description of the layout of this regions.

CONFIGURATION SPACE REGISTERS

Each PCI function implements configuration space as defined in PCI Bus Local Specification, Revision 2.2 (Revision 2.3 for the BCM4401 B0 and later). These registers can be accessed only with PCI configuration reads and writes. Configuration space is not accessible from the system backplane. The host software must use one of the SBtoPCITranslation registers (see "Address Translation Registers" on page 159) to map a portion of the backplane address space to type 0 configuration accesses at an address that would generate an IDSEL to this device.

In addition to the standard set of PCI configuration registers, the PCI core supplies a power management capability blocks and a set of extension registers. For a complete description of the standard PCI registers, refer to the PCI specification.

The configuration space registers include the following register sets:

STANDARD CONFIGURATION REGISTERS

The following registers are used for standard configuration:

VENDOR ID REGISTER (VENDORID, OFFSET 0X00)

The two-byte, read-only PCI Vendor ID register always contains the value 0x14E4.

Table 147: Vendor ID Register Bit Field Descriptions

Field	Name	Description	Access	Default
15:0	VendorID	PCI Vendor ID	RO	0x14E4

DEVICE ID REGISTER (DEVICEID, OFFSET 0X02)

The two-byte, read-only Device ID register contains a chip-specific value that is read from the SPROM on device reset. See Table 163: "SPROM Layout (BCM4401 B0 and later)," on page 206 for more details.

Table 148: Device ID Register Bit Field Descriptions

Field	Name	Description	Access	Default
15:0	DeviceID	BCM4401 AX	RO	0x4401
15:0	DeviceID	BCM4402	RO	0x4402
15:0	DeviceID	BCM4401 B0 and later	RO	0x170c

COMMAND REGISTER (COMMAND, OFFSET 0X04)

The two-byte Command register is both readable and writable and contains the value 0x0 at PCI reset.

Table 149: Command Register Bit Field Descriptions

Field	Name	Description	Access	Default
15:11	Reserved	–	R/W	0
10	Reserved	–	R/W	0
10	Interrupt Disable (ID) BCM4401 B0 only	This read/write field is set to 1 to disable assertion of PCI interrupts and set to 0 to enable interrupts. The contents of this field do not affect the value reported in the InterruptStatus field of the Status register (see "Status Register (Status, offset 0x06)" on page 167).	R/W	0
9	FbtbEn (FE)	Fast Back-to-Back Enable This field is fixed at 0, indicating that the device does not support fast back-to-back transactions as a bus master even if the target devices do.	R/W	0
8	SerrEn (SE)	SERR Enable This read/write field is set to 1 to enable assertions of the PCI_SERR_L signal and set to 0 to disable use of that signal.	R/W	0
7	AddrStepCtrl (AC)	Address Step Control This field is fixed at 0, indicating that the device does not support PCI address and data stepping.	R/W	0
6	PER (PR)	Parity Error Response This read/write field is set to 1 to enable reporting of parity errors and set to 0 to ignore parity errors.	R/W	0
5	VGA (VG)	VGA Palette Snoop This field is fixed at 0; the device does not special-case VGA palette accesses.	R/W	0
4	MemWrInv (WI)	Memory Write/Invalidate This field is fixed at 0, indicating that the device does not support the PCI memory write and invalidate when acting as a bus master.	R/W	0
3	SpecialCycEn (SP)	Special Cycle Enable This field is fixed at 0, indicating that the device does not respond to PCI special cycle accesses.	R/W	0
2	BusMasterEn (ME)	Bus Master Enable This read/write field is set to 1 to enable the device to act as a PCI bus master and set to 0 otherwise.	R/W	0
1	Mem (MM)	Memory Space This read/write field is set to 1 to enable responses to PCI memory space accesses and set to 0 to disable responses to those accesses.	R/W	0
	Mem (MM) BCM4401 B0 only	Memory Space. This read/write field is set to 1 to enable response to PCI memory space accesses and set to 0 to disable responses to those accesses. This field is not writable when the SPROM BAR0IO field (see Table 163: "SPROM Layout (BCM4401 B0 and later)," on page 206) is set to 1.	R/W	0

Table 149: Command Register Bit Field Descriptions (Cont.)

Field	Name	Description	Access	Default
0	IO (IO)	I/O Space This field is fixed at 0, indicating that the device does not respond to PCI I/O space accesses.	R/W	0
	IO (IO) BCM4401 B0 only	I/O Space This read/write field is set to 1 to enable response to PCI I/O space accesses and set to 0 to disable responses to those accesses. This field is not writable when the SPROM BAR0IO field (see Table 163: "SPROM Layout (BCM4401 B0 and later)," on page 206) is set to 0.	R/W	0

STATUS REGISTER (STATUS, OFFSET 0X06)

The two-byte Status register is both readable and writable and contains the value 0x10 at PCI reset. The writable fields in this register can only be set by hardware and can be cleared by writing them with a 1. Writing a 0 to a field has no effect.

Table 150: Status Register Bit Field Descriptions

Field	Name	Description	Access	Default
15	PTY (PE)	Parity Error Detected This field is set to 1 when the device detects a parity error. Writing a 1 to this field clears the field.	R/W	–
14	SERR (SR)	SERR Asserted This field is set to 1 when the device asserts the PCI_SERR_L signal. Writing a 1 to this field clears the field.	R/W	–
13	MABT (MA)	Master Abort This field is set to 1 when the device terminates its own transaction with a master abort. Writing a 1 to this field clears the field.	R/W	–
12	RABT (RT)	Received Target Abort This field is set to 1 when another device terminates a transaction started by the device with a target abort. Writing a 1 to this field clears the field.	R/W	–
11	SABT (ST)	Signaled Target Abort This field is set to 1 when the device terminates a transaction with a target abort. Writing a 1 to this field clears the field.	R/W	–
10:9	DEVSEL (DS)	DEVSEL Timing This field is fixed at the value 0x0, indicating that the device uses the fast device select assertion time.	R/W	–
8	PERR (PR)	Data Parity Error Detected This field is set to 1 when the detected or asserts PCI_PERR_L when the PER bit in the command register is set. Writing a 1 to this field clears the field.	R/W	–
7	FBTB (BB)	Fast Back-to-Back Capable This field is fixed at 0, indicating that the device does not support fast back-to-back transactions to different agents.	R/W	–
6	Reserved		R/W	–
5	66MHz (66)	66MHz Capable This field is fixed at 0, indicating that the device does not support a 66 MHz PCI bus.	R/W	–
4	Cap (CP)	Capabilities Defined This field is fixed at 1, indicating that the device supports PCI extended capabilities. The device supports the PCI power management capability.	R/W	–
3	Reserved	–	R/W	–
3 4401B0 only	Interrupt Status (IS)	This field is set to 1 when this function has a PCI interrupt pending and set to 0 when no interrupts are pending. A pending interrupt causes an assertion of PCI INTA# unless the InterruptDisable field of the Command register is set ("Command Register (Command, offset 0x04)" on page 165).	R/O	0
2:0	Reserved	–	R/W	–

REVISION ID REGISTER (REVISIONID, OFFSET 0X08)

The one-byte, read-only Revision ID register identifies the specific revision of this adapter.

CLASS CODE REGISTER (CLASSCODE, OFFSET 0X09)

The three-byte, read-only Class Code register always contains the value 0x020000.

LATENCY TIMER REGISTER (LATENCY, OFFSET 0X0D)

The one-byte Latency Timer register can be programmed to expire in a period from 0 to 255 PCI bus clocks.

CONFIGURATION HEADER TYPE REGISTER (HEADERTYPE, OFFSET 0X0E)

The one-byte, read-only Configuration Header Type register contains the value 0x0 for functions other than Function 0 and contains the value 0x80 when reading Function 0 configuration space.

BASE ADDRESS 0 REGISTER (BAR0, OFFSET 0X10)

(BCM4401B0 only) The BAR0 register can be configured to map I/O space or memory space depending on the settings in the SPROM (see Table 163: "SPROM Layout (BCM4401 B0 and later)," on page 206). When BAR0 is configured to map into I/O space (the BAR0IO bit in the SPROM is set to 1), the register maps a 256 byte I/O region. When BAR0 is configured to map into memory space (the BAR0IO bit in the SPROM is set to 0), BAR0 maps an 8KB memory region consisting of a 4KB window into the backplane space used for core accesses, followed by a 4KB windows used for direct accesses to the SPROM and the PCI host interface registers.

Table 151: Base Address 0 Register Memory Mapped Bit Field Descriptions (BCM4401 B0 Only)

Field	Name	Description	Access	Default
31:8	BaseAddr (BA)	Base Address his field should be initialized with a base address of the I/O range allocated to this device	R/W	0
7:1	Reserved	–	R/O	0
0	OSpace (IS)	The field is fixed to a 1, indicating that the recognized address range lies in I/O space.	R/O	1

The Base Address 0 register (BAR0) is both readable and writable and contains the value 0x0 at device reset. When enabled, BAR0 maps an 8KB memory region consisting of a 4KB window into the backplane space used for core accesses, followed by a 4KB window used for direct accesses to the SPROM and the PCI host interface registers.

- 4 KB: Small window into system backplane space, intended for access to cores

Table 152: Base Address 0 Register Memory Mapped Bit Field Descriptions

Field	Name	Description	Access	Default
31:13	BaseAddr (BA)	Base Address This field should be initialized with the base address of the memory range allocated to this device.	R/W	0
12:4	Rsvd	Reserved This field is reserved and always contains all zeros, ensuring the memory range allocated to the device is aligned on a 4 KB boundary.	R/O	0
3	PrefetchEn (PE)	Prefetch Enable This field is fixed at 0, indicating that memory references are not prefetchable and writes to this address range cannot be merged.	R/O	0
2:1	Type (TP)	Memory Type This field is fixed at 0, indicating that the address range can be relocated anywhere within the 32-bit memory space.	R/O	0
0	MemSpace (MS)	Memory Space Indicator This field is fixed at 0, indicating that the recognized address range lies in memory space.	R/O	0

BASE ADDRESS 1 REGISTER (BAR1, OFFSET 0X14)

The Base Address 1 register is both readable and writable and contains the value 0x0 at device reset.

- Large window into system backplane space, intended for memory

Table 153: Base Address 1 Register Bit Field Descriptions

Field	*Name*	*Description*	*Access*	*Default*
31:27/ 31:20	BaseAddr (BA)	Base Address This field should be initialized with the base address of the memory range allocated to this device.	R/W	0
26:4/ 19:4	Rsvd	Reserved This field is reserved and always contains all zeros, ensuring the memory range allocated to the device is aligned on a 4 KB boundary.	R/W	0
3	PrefetchEn (PE)	Prefetch Enable This field is fixed at 0, indicating that memory references are not prefetchable and writes to this address range cannot be merged.	R/W	0
2:1	Type (TP)	Memory Type This field is fixed at 0, indicating that the address range can be relocated anywhere within the 32-bit memory space.	R/W	0
0	MemSpace (MS)	Memory Space Indicator This field is fixed at 0, indicating that the recognized address range lies in memory space.	R/W	0

PCI Subsystem Vendor ID Register (SubsystemVendorID, offset 0x2C)

The two-byte, read-only PCI Subsystem Vendor ID register is read from the SPROM on device reset. See Table 163: "SPROM Layout (BCM4401 B0 and later)," on page 206 for more details.

PCI Subsystem ID Register (SubsystemID, offset 0x2E)

The two-byte, read-only PCI Subsystem ID register is read from the SPROM on device reset. See Table 163: "SPROM Layout (BCM4401 B0 and later)," on page 206 for more details.

Expansion ROM Bar Register (ExpansionROMBar, offset 0x30)

The four-byte, read-only Expansion ROM Bar register is both readable and writable and contains the value 0x0 at PCI reset.

Table 154: Expansion ROM Bar Register Bit Field Descriptions

Field	Name	Description	Access	Default
31:10	BaseAddr (BA)	Base Address This field should be initialized with the base address of the memory range allocated to the expansion ROM.	R/W	0
9:1	Reserved	–	R/W	0
0	Enable (EN)	Enable When this field is set to 1, accesses to expansion ROM space are permitted. When this field is cleared to 0, accesses to expansion ROM space are not claimed.	R/W	0

Capability List Pointer Register (CapPtr, offset 0x34)

The one-byte, read-only Capability List Pointer register contains the eight-bit offset in configuration space of the first extended capability definition block. This register always contains the value 0x40.

PCI Interrupt Line Register (IntLine, offset 0x3C)

The one-byte PCI Interrupt Line register is both readable and writable and is used by device drivers.

PCI Interrupt Pin Register (IntPin, offset 0x3D)

The one-byte PCI Interrupt Pin register is fixed at the value 0x1, indicating that the device always uses the PCI_INT_L interrupt pin.

PCI Minimum Grant Register (MinGnt, offset 0x3E)

The one-byte PCI Minimum Grant register is both readable and writable and contains the value 0x0 at device reset.

PCI Maximum Latency Register (MaxLat, offset 0x3F)

The one-byte PCI Maximum Latency register is both readable and writable and contains the value 0x0 at device reset.

POWER MANAGEMENT CAPABILITY DEFINITION BLOCK

This section defines the extended capability block used for power management. This block is located at offset 0x40 in configuration space.

The following registers are used for power management:

- "PCI Power Management Capability ID Register (CapID, offset 0x40)"
- "PCI Next Capability Pointer Register (CapNext, offset 0x41)"
- "PCI Power Management Capability Register (PMC, offset 0x42)" on page 173
- "PCI Power Management Control/Status Register (PMCSR, offset 0x44)" on page 174
- "PCI Power Management Data Register (PMData, offset 0x48)" on page 175

PCI POWER MANAGEMENT CAPABILITY ID REGISTER (CAPID, OFFSET 0X40)

This one-byte, read-only PCI Power Management Capability ID register contains the value 0x1, indicating that this capability block contains PCI power management registers.

PCI NEXT CAPABILITY POINTER REGISTER (CAPNEXT, OFFSET 0X41)

The one-byte, read-only PCI Next Capability Pointer register contains the value 0x0, indicating no more defined capability blocks.

PCI POWER MANAGEMENT CAPABILITY REGISTER (PMC, OFFSET 0X42)

The two-byte, read-only PCI Power Management Capability register contains information on device support for PCI power management.

Table 155: PCI Power Management Capability Register Bit Field Descriptions

Field	*Name*	*Description*	*Access*	*Default*
15:11	PME (PM)	Power Management Event Support This field indicates the power states in which the device can assert a power management event.	RO	–
10	D2 (D2)	D2 Support This field contains the value 0x1, indicating that the device supports the D2 power state.	RO	–
9	D1 (D1)	D1 Support This field contains the value 0x1, indicating that the device supports the D1 power state.	RO	–
8:6	AuxCurr (AX)	Auxiliary Current This field always contains the value 0x7, indicating that the device requires 375 mA of current from the auxiliary power source.	RO	–
5	DSI (DS)	Device-Specific Initialization This field contains the value 0x0, indicating that no device-specific initialization is required upon entrance to the D0 state before the driver can access the chip.	RO	–
4	Reserved	–	RO	–
3	PmeClk (PC)	PME Clock Required This field contains the value 0x0, indicating that the device can deliver power management events in the absence of a PCI clock.	RO	–
2:0	VerID (VR)	Version ID This field contains the value 0x2, indicating compliance with version 1.1 of the specification.	RO	–

PCI POWER MANAGEMENT CONTROL/STATUS REGISTER (PMCSR, OFFSET 0X44)

The two-byte PCI Power Management Control/Status register is both readable and writable and contains control and status information on device support for PCI Power management. Because the device can deliver power management events from the $D3_{cold}$ state, the value in this register is not changed by device reset. In particular, the PmeEn and PmeStat fields are undefined at initial power on and reflect PME status when the device is awakened from the $D3_{cold}$ power state. The PMEStatus field of the PMCSR register cannot be cleared without first clearing the source of the PME in the core that signaled the event.

Table 156: PCI Power Management Control/Status Register Bit Field Descriptions

Field	Name	Description	Access	Default
15	PmeStat (ST)	PME Status This field is set to 1 when a power management event is pending, regardless of the setting of the PmeEn field. Host driver software should write a 1 to this field to clear a pending event; writing a 0 to this field has no effect.	R/W	–
14:13	DataScale (DL)	Data Scale This field contains the value 0x2, indicating that values returned in the PMData register are expressed in 10-mW units.	R/W	–
12:9	DataSel (DS)	Data Select This field selects the data item reported by reads to the PMData register. See Table 157 on page 174 for a description of the data items.	R/W	–
8	PmeEn (PE)	PME Enable This field should be set to enable the assertion of power management events and clear to disable the assertion of those events.	R/W	–
7:2	Reserved	–	R/W	–
1:0	PowState (PS)	Power State This field contains the current power state of the device. Altering this field changes the current power state. Table 158 on page 175 contains the definitions for this field.	R/W	–

Table 157: Power Management Data Items

Data Item	Description
0	D0 power consumed
1	D1 power consumed
2	D2 power consumed
3	D3 power consumed
4	D0 power dissipated
5	D1 power dissipated
6	D2 power dissipated
7	D3 power dissipated
8	D1 Common logic power consumed (iLine32 function only)
9-0xf	Reserved

Table 158: Power State Encoding

Power State	*Value*
D0	0x0
D1	0x1
D2	0x2
$D3_{hot}$	0x3

PCI Power Management Data Register (PMData, offset 0x48)

The one-byte, read-only PCI Power Data register is used to read the power consumption and power dissipation values selected by the DataSel field of the PMCSR register (see "PCI Power Management Control/Status Register (PMCSR, offset 0x44)" on page 174).

Device Specific Configuration Space Registers

The following registers are used for power management:

- "PCI to Bar 0 Window Register (PCIBar0Window, offset 0x80)" on page 176
- "PCI to Bar 1 Window Register (PCIBar1Window, offset 0x84)" on page 176
- "SPROM Control Register (SPROMControl, Offset 0x88)" on page 177
- "Bar 1 Burst Control Register (BAR1BurstControl, offset 0x8c)" on page 178
- "PCI Interrupt Status Register (PCIIntStatus, offset 0x90)" on page 179
- "PCI Interrupt Mask Register (PCIIntMask, offset 0x94)" on page 180
- "PCI to System Backplane Mailbox Register (PCItoSBMailbox, offset 0x98)" on page 181

PCI TO BAR 0 WINDOW REGISTER (PCIBAR0WINDOW, OFFSET 0X80)

The four-byte PCI Bar 0 Window register is both readable and writable and contains 0x0 at PCI reset. The contents of this register specify the region of backplane address space that is accessed by references mapped by the BAR0 register (see "Base Address 0 Register (BAR0, offset 0x10)" on page 169).

Bits [31:12] ([31:8] for BCM4401 B0 only) of PCI addresses that match the lower 4 KB of BAR0 are replaced with bits [31:12] ([31:8] for BCM4401 B0 only) of this register to form the address for the system backplane request generated by the matching access.

This version of the PCI to Bar 0 Window register applies to the BCM4401 B0 and later controller only.

Table 159: PCI to Bar 0 Window Register Bit Field Descriptions (BCM4401 B0 and Later)

Field	Name	Description	Access	Default
31:8	–	–	R/W	0
7:0	–	–	R/O	0

This version of the PCI to Bar 0 Window register applies to the rest of the BCM440X family.

Table 160: PCI to Bar 0 Window Register Bit Field Descriptions (Rest of BCM440X Family)

Field	Name	Description	Access	Default
31:12	–	–	R/W	0
11:0	–	–	R/O	0

PCI TO BAR 1 WINDOW REGISTER (PCIBAR1WINDOW, OFFSET 0X84)

The four-byte PCI Bar 1 Window register is both readable and writable and contains the value 0x0 at PCI reset. The contents of this register specify the region of backplane address space that is accessed by references mapped by the BAR1 register (see "Base Address 1 Register (BAR1, offset 0x14)" on page 170).

SPROM CONTROL REGISTER (SPROMCONTROL, OFFSET 0X88)

The four-byte SPROM Control register is both readable and writable and contains the value 0x0 at PCI reset.

Table 161: SPROM Control Register Bit Field Descriptions

Field	Name	Description	Access	Default
31:7	Reserved	–	R/O	0
6	Reserved	–	R/O	0
	BPAccessEn (BE) (BCM4401 B0 only)	Backplane Access Enable. When this field is set to 1, accesses to the BackplaneData register (see "Backplane Data Register (BackplaneData, offset 0xa4)" on page 184) are forwarded to the system backplane. When this field is set to 0, writes to the BackplaneData register do not go to the backplane and reads of the BackplaneData register return the contents of the last value written.	R/W	0
5	BootROMWriteEn (BW)	Boot ROM Write Enable. When this field is set to 1, the contents of the external Boot ROM can be modified. When this field is set to 0, writes to the Boot ROM cause target aborts.	R/W	0
4	SPROMWriteEn (SW)	SPROM Write Enable. When this field is set to 1, the contents of the external SPROM can be modified using writes to the upper 4KB of the BAR0 address space. When this field is set to 0, writes to that region are ignored. The contents of this field are ignored if the SPROMLocked field is set.	R/W	0
3	SPROMLocked (SL)	SPROM Locked. This field is set to 1 to indicate that the SPROM is locked and cannot be modified, as indicated by the device strapping options.	R/O	–
2	SPROMBlank (SB)	SPROM Blank. This field is set to 1 to indicate that the SPROM is blank, as indicated by device strapping options.	R/O	–
1:0	SPROMSize (SS)	SPROM Size. This field reflects the size of the external SPROM. The following values are supported: 00 - 1 KB 01 - 4 KB 10 - 16 KB 11 - Reserved	R/O	–

BAR 1 Burst Control Register (BAR1BurstControl, offset 0x8C)

The four-byte Base Address 1 Burst Control register is both readable and writable and contains the value 0x0 at PCI reset.

The BistEnable, BISTDone, and BistFail fields enable BIST to be executed from the PCI bus. The BistEnable field of the PCI core's SBTMSTATELOW register (see "System Backplane Target State Low Register (SBTMSTATELOW, offset 0xF98)" on page 17) should be used to execute BIST from an embedded processor.

Table 162: BAR 1 Burst Control Register Bit Field Descriptions

Field	Name	Description	Access	Default
31:5	Reserved	–	R/W	0
4	TransactionTimerEn (ET)	Transaction Timer Enable When this field is set to 1, a 256 PCI cycle transaction timer is enabled. This timer is used to terminate bus transfers that are claimed by targets that fail to respond with a data phase in the maximum amount of time dictated by the PCI specification.	R/W	0
	BISTDone (BD) BCM4401 B0 only	This field is set to zero when the BISTEnable field is set to 1 and this field is set to 1 when the core's built-in self-test completes.	R/W	0
3	BISTFail (BF)	BIST Fail This field is set to 1 when the core's BIST (Built-In Self-Test) detects an error. This field is valid after the BISTEnable field (BISTDone for BCM4401 B0 only) is set to one. The BistStatus register (see "Built-In Self Test Register (BISTStatus, offset 0x0C)" on page 47) in the core contains more information on the failure.	R/W	0
2	BISTEnable (BE)	BIST Enable The contents of this field are combined with the BISTEnable field of the PCI core's SBTMSTATELOW register (see "System Backplane Target State Low Register (SBTMSTATELOW, offset 0xF98)" on page 17). When either field is set to 1, the core initiates the BIST (Built-In Self-Test) cycle for all RAMs in this core. After writing this field to 1, software should wait until the BISTDone field contains 1 to read the BISTFail field, and clear this field to 0.	R/W	0
1	PrefetchEn (PE)	Prefetch Enable When this field is set to 1, PCI reads in the BAR1 region cause aligned four-word bursts to be prefetched into a local buffer and made available for PCI bursts. When cleared to 0, reads in the BAR1 region are terminated after a single data phase. Setting this field improves performance. Reads in the BAR0 region are not affected by this field.	R/W	0
0	WriteBufferEn (WB)	Write Buffer Enable When this field is set to 1, PCI writes bursts to the BAR1 region are accepted. When cleared to 0, PCI write bursts are terminated after a single data phase. Setting this field improves performance. Writes to the BAR0 region are not affected by this field.	R/W	0

PCI INTERRUPT STATUS REGISTER (PCIINTSTATUS, OFFSET 0X90)

The four-byte PCI Interrupt Status register is both readable and writable and contains the value 0x0 at PCI reset. This register contains both indications of interrupt-causing events generated by the PCI core and the interrupt output from all other cores. These events cause PCI interrupts when the corresponding field in the PCIIntMask register (see "PCI Interrupt Mask Register (PCIIntMask, offset 0x94)" on page 180) is also set and the InterruptDisable field of the Command register (see Table : "Command Register (Command, offset 0x04)," on page 165) is set to 0.

This version of the PCI Interrupt Status register applies to the BCM4401 B0 and later controller only.

Table 163: PCI Interrupt Status Register Bit Field Descriptions (BCM4401 B0 and Later)

Field	Name	Description	Access	Default
31:16	Reserved	–	R/W	0
15:8	SBIntMask (IM)	This field contains one bit for each of the 8 backplane flags that are used for interrupts. The mapping of cores to flags is device specific.	R/W	0
7:3	Reserved	–	R/W	0
2	SBError (BE)	System Backplane Error This field is set to 1 when an assertion of the system backplane error signal is detected. This field is cleared to 0 by writing a 1 to it.	R/W	0
1:0	SBtoPCIMailbox (SM)	System Backplane to PCI Mailbox This field is set by writing the SBtoPCIMailbox register field for this function. The bits in this field are cleared by writing ones to them.	R/W	0

This version of the PCI Interrupt Status register applies to the rest of the BCM440X family.

Table 164: PCI Interrupt Status Register Bit Field Descriptions (rest of the BCM440X family)

Field	Name	Description	Access	Default
31:3	Reserved	–	R/W	0
2	SBError (BE)	System Backplane Error This field is set to 1 when an assertion of the system backplane error signal is detected. This field is cleared to 0 by writing a 1 to it.	R/W	0
1:0	SBtoPCIMailbox (SM)	System Backplane to PCI Mailbox This field is set by writing the SBtoPCIMailbox register field for this function. The bits in this field are cleared by writing ones to them.	R/W	0

PCI INTERRUPT MASK REGISTER (PCIINTMASK, OFFSET 0X94)

The 4-byte PCI Interrupt Mask register is both readable and writable and contains a device-dependent value at PCI reset. This register enables the generation of PCI interrupts from PCI core-detected events and from the interrupt outputs of all other cores. These events cause PCI interrupts when the corresponding field in the PCIIntStatus register (see "PCI Interrupt Status Register (PCIIntStatus, offset 0x90)" on page 179) is set to 0.

This version of the PCI Interrupt Mask register applies to the BCM4401 B0 and later controller only.

Table 165: PCI Interrupt Mask Register Bit Field Descriptions (BCM4401 B0 and Later)

Field	Name	Description	Access	Default
31:16	Reserved	–	R/W	0
15:8	SBIntMask (IM)	This field contains one bit for each of the 8 backplane flags that are used for interrupts. The mapping of cores to flags is device specific. At PCI reset, this field is initialized so that the interrupt for the core whose enumeration space is addressed by the PciBAR0Window register is enabled, and all other core interrupts are disabled. If that register has not been initialized in the SPROM, this field contains 0 at device reset.	R/W	0
7:3	Reserved	–	R/W	0
2	SBError (SB)	System Backplane Error When set to 1, this field enables interrupts caused by assertions of the system backplane error signal.	R/W	0
1:0	SBMailbox (BM)	System Backplane Mailbox When set to 1, bits in this field enable interrupts caused by writes to the SBtoPCIMailbox register (see "System Backplane to PCI Mailbox Register (SBtoPCIMailbox, offset 0x28)" on page 155) for this function.	R/W	0

This version of the PCI Interrupt Mask register applies to the rest of the BCM440X family.

Table 166: PCI Interrupt Mask Register Bit Field Descriptions (rest of the BCM440X family)

Field	Name	Description	Access	Default
31:3	Reserved	–	R/W	0
2	SBError (SB)	System Backplane Error When set to 1, this field enables interrupts caused by assertions of the system backplane error signal.	R/W	0
1:0	SBMailbox (BM)	System Backplane Mailbox When set to 1, bits in this field enable interrupts caused by writes to the SBtoPCIMailbox register (see "System Backplane to PCI Mailbox Register (SBtoPCIMailbox, offset 0x28)" on page 155) for this function.	R/W	0

PCI TO SYSTEM BACKPLANE MAILBOX REGISTER (PCITOSBMAILBOX, OFFSET 0X98)

The four-byte PCI to System Backplane Mailbox register is write-only. This register is used to signal backplane interrupts. Fields in this register are cleared when they are read from the core's IntStatus register (see "Interrupt Status Register (IntStatus, offset 0x20)" on page 153).

Table 167: PCI to System Backplane Mailbox Register Bit Field Descriptions

Field	Name	Description	Access	Default
31:2	Reserved	–	WO	–
1:0	SBMailbox (BM)	System Backplane Mailbox Bits in this field are set to 1 to signal backplane interrupts.	WO	–

BACKPLANE ACCESS REGISTERS

The BackplaneAddress and BackplaneData registers allow accesses to arbitrary addresses on the system backplane to be performed through PCI configuration space.

The following registers are used for backplane access:

- "Backplane Address (BackplaneAddress, offset 0xa0)" on page 183
- "Backplane Data Register (BackplaneData, offset 0xa4)" on page 184

BACKPLANE ADDRESS (BACKPLANEADDRESS, OFFSET 0XA0)

The four-byte Backplane Address register is both readable and writable and contains the value 0x1000 at device reset. The contents of this register are used to address an arbitrary location on the system backplane.

This version of the Backplane Address applies to the BCM4401 B0 and later controller only.

Table 168: Backplane Address Register Bit Field Descriptions (BCM4401 B0 and Later)

Field	Name	Description	Access	Default
31	BootROM (BR)	When this field is set to 1, accesses to the BackplaneData register (see "Backplane Data Register (BackplaneData, offset 0xa4)" on page 184) are directed to the external expansion ROM. When this field is set to 0, BackplaneData accesses are directed to the backplane.	R/W	0
30:17	Reserved	–	R/O	0
16:0	Offset (OS)	Offset When the BootROM field is set to 1, the contents of this field addresses an offset in the Expansion ROM. If both BootROM and Offset[12] are clear, bits [31:12] of the PCI Bar0Window register are concatenated with the contents of bits [11:0] of this field to form the system backplane address used when the BackplaneData register is accessed. If the BootROM bit is clear and Offset[12] is set, the contents of bits [11:0] are used to address SPROM shadow space.	R/W	0

This version of the Backplane Address applies to the rest of the BCM440X family.

Table 169: Backplane Address Register Bit Field Descriptions (rest of the BCM440X family)

Field	Name	Description	Access	Default
31:13	–	–	R/W	–
12:0	Offset (OS)	Offset If Offset[12] is set to 0, bits [31:12] of the PCIBAR0Window register (see "PCI to Bar 0 Window Register (PCIBar0Window, offset 0x80)" on page 176) are concatenated with the contents of bits [11:0] of this field to form the system backplane address used when the BackplaneData register (see "Backplane Data Register (BackplaneData, offset 0xa4)" on page 184) is accessed. If Offset[12] is set to 1, the contents of bits (11:0) are used to address SPROM shadow space when the BackplaneData register is accessed.	R/W	–

BACKPLANE DATA REGISTER (BACKPLANEDATA, OFFSET 0XA4)

The four-byte Backplane Data register is both readable and writable and contains an undefined value at device reset. When the BPAccessEn field of the SPROMControl register is set to 1 (see "SPROM Control Register (SPROMControl, Offset 0x88)" on page 177), reads to this register return the data at the location specified by the contents of the BackplaneAddress register (see "Backplane Address (BackplaneAddress, offset 0xa0)" on page 183), and writes to this register modify the contents of that location. When backplane accesses are disabled, writes to this register are not propagated to the backplane and reads return the last value written.

Table 170: Backplane Data Register Bit Field Descriptions

Field	*Name*	*Description*	*Access*	*Default*
31:0	BackplaneData	Backplane Data	R/W	Not defined

UNIMPLEMENTED PCI CONFIGURATION REGISTERS

The device does not implement the PCI Cache Line Size, BAR2-5, MinGnt, and MaxLat registers. The PMCSR Bridge Support Extensions register in the Power Management Capability Block is also not implemented.

Section 7: Programming Hints

This chapter provides additional information for software developers on common operations affecting the entire BCM440X.

RESET

After the BCM440X has been powered on, the only way to reset the entire chip is the PCI reset function.

Individual cores can be reset independently. The external interface SDRAM controller, and PCI cores should not be reset when there are outstanding transactions that access their memory spaces. The mechanism for resetting and restarting cores is common across all cores and consists of several accesses that manipulate the Reset, Reject, ClkEnable, and ForceGatedClocksOn fields of the SBTMSTATELOW register (see "System Backplane Target State Low Register (SBTMSTATELOW, offset 0xF98)" on page 17) in the core's enumeration address space. Not all cores support gated clocks; for cores that do not, changes to the ForceGatedClocksOn field are simply ignored.

When modifying the reset field, software must use the ForceGatedClocksOn and ClkEnable fields to force reset to propagate throughout the entire core. Failure to follow the recommended sequences could result in the core starting with inconsistent state or powering down while driving invalid outputs.

A running core must be shut down cleanly before it can be reinitialized. At power-on-reset time or after restarting from a PCI power down state, only the initialization sequence is necessary. In all other cases, the steps listed for disabling a core should be executed before the initialization steps.

The following sequence of three writes to SBTMSTATELOW initializes the core correctly. Wait for 1 us between each write to allow state to propagate.

1. Write 0x30001, asserting reset while enabling the clocks and forcing them on throughout the core.
2. Write 0x30000, clearing reset and allowing it to propagate throughout the core.
3. Write 0x10000, leaving clocks enabled but allowing them to be turned off for inactivity.

Some cores require additional SBTMSTATELOW fields to be asserted during the initialization sequence. See the documentation of individual cores for more information on reset requirements.

The following sequence of register accesses disables the core correctly. This sequence should be executed only if the core has previously been initialized.

1. If the core has a DMA processor, disable all transmit and receive channels.
2. Write 0x10002 to SBTMSTATELOW, setting the Reject field in order to ensure that no further transactions reach this core.
3. Read that register back until the Reject field is observed to be set.
4. Read the SBTMSTATEHIGH register (see "System Backplane Target State High Register (SBTMSTATEHIGH, offset 0xF9C)" on page 18) until the Busy field is observed to be clear. This ensures that all transactions initiated by this core have completed.
5. Write 0x30003 to SBTMSTATELOW, asserting reset and reject while enabling the clocks and forcing them on throughout the core.
6. Write 0x3 to SBTMSTATELOW, leaving reset and reject asserted while disabling the clocks. This write should be separated from the previous write by a minimum of 1 µs.

INTERNAL CLOCK FREQUENCIES

The BCM440X contains a PLL that produces a 62.5-MHz clock based on a 25-MHz oscillator input. The 62.5-MHz clock is used for the system backplane and the Ethernet MAC and Codec cores. An externally generated PCI bus clock must be connected to the PCI_CLK pin.

CALCULATING THE CRC8 CHECKSUM

This section details the algorithm used by the BCM440X to calculate a Cyclic Redundancy Check (CRC) on the SPROM contents. Unlike some CRC algorithms that use a polynomial, the BCM440X uses a lookup table.

```
typedef unsigned char u8;
typedef unsigned long u32;

#define CRC_INNER_LOOP(n, c, x) (c) = ((c) >> 8) ^ crc8_table[((c) ^(x)) & 0xff]

static u8 crc8_table[256] = {
0x00, 0xF7, 0xB9, 0x4E, 0x25, 0xD2, 0x9C, 0x6B,
0x4A, 0xBD, 0xF3, 0x04, 0x6F, 0x98, 0xD6, 0x21,
0x94, 0x63, 0x2D, 0xDA, 0xB1, 0x46, 0x08, 0xFF,
0xDE, 0x29, 0x67, 0x90, 0xFB, 0x0C, 0x42, 0xB5,
0x7F, 0x88, 0xC6, 0x31, 0x5A, 0xAD, 0xE3, 0x14,
0x35, 0xC2, 0x8C, 0x7B, 0x10, 0xE7, 0xA9, 0x5E,
0xEB, 0x1C, 0x52, 0xA5, 0xCE, 0x39, 0x77, 0x80,
0xA1, 0x56, 0x18, 0xEF, 0x84, 0x73, 0x3D, 0xCA,
0xFE, 0x09, 0x47, 0xB0, 0xDB, 0x2C, 0x62, 0x95,
0xB4, 0x43, 0x0D, 0xFA, 0x91, 0x66, 0x28, 0xDF,
0x6A, 0x9D, 0xD3, 0x24, 0x4F, 0xB8, 0xF6, 0x01,
0x20, 0xD7, 0x99, 0x6E, 0x05, 0xF2, 0xBC, 0x4B,
0x81, 0x76, 0x38, 0xCF, 0xA4, 0x53, 0x1D, 0xEA,
0xCB, 0x3C, 0x72, 0x85, 0xEE, 0x19, 0x57, 0xA0,
0x15, 0xE2, 0xAC, 0x5B, 0x30, 0xC7, 0x89, 0x7E,
0x5F, 0xA8, 0xE6, 0x11, 0x7A, 0x8D, 0xC3, 0x34,
0xAB, 0x5C, 0x12, 0xE5, 0x8E, 0x79, 0x37, 0xC0,
0xE1, 0x16, 0x58, 0xAF, 0xC4, 0x33, 0x7D, 0x8A,
0x3F, 0xC8, 0x86, 0x71, 0x1A, 0xED, 0xA3, 0x54,
0x75, 0x82, 0xCC, 0x3B, 0x50, 0xA7, 0xE9, 0x1E,
0xD4, 0x23, 0x6D, 0x9A, 0xF1, 0x06, 0x48, 0xBF,
0x9E, 0x69, 0x27, 0xD0, 0xBB, 0x4C, 0x02, 0xF5,
0x40, 0xB7, 0xF9, 0x0E, 0x65, 0x92, 0xDC, 0x2B,
0x0A, 0xFD, 0xB3, 0x44, 0x2F, 0xD8, 0x96, 0x61,
0x55, 0xA2, 0xEC, 0x1B, 0x70, 0x87, 0xC9, 0x3E,
0x1F, 0xE8, 0xA6, 0x51, 0x3A, 0xCD, 0x83, 0x74,
0xC1, 0x36, 0x78, 0x8F, 0xE4, 0x13, 0x5D, 0xAA,
0x8B, 0x7C, 0x32, 0xC5, 0xAE, 0x59, 0x17, 0xE0,
0x2A, 0xDD, 0x93, 0x64, 0x0F, 0xF8, 0xB6, 0x41,
0x60, 0x97, 0xD9, 0x2E, 0x45, 0xB2, 0xFC, 0x0B,
0xBE, 0x49, 0x07, 0xF0, 0x9B, 0x6C, 0x22, 0xD5,
0xF4, 0x03, 0x4D, 0xBA, 0xD1, 0x26, 0x68, 0x9F
};
```

```
/* The argument pdata is a pointer to the SPROM image in RAM */
u8 crc8(u8 *pdata)
    {
    u32 nbytes = 128; /* SPROM size minus 1 */
    u8 crc = 0xFF;

    while (nbytes-- > 0)
    CRC_INNER_LOOP(8, crc, *pdata++);

    return crc;
    }
```

SPROM LAYOUT

Table 171: SPROM Layout (BCM4401 B0 and Later)

BYTES	15 [F]	14 [E]	13 [D]	12 [C]	11 [B]	10 [A]	09	08	07	06	05	04	03	02	01	00
1:0 [1:0]	Bits [15:12] of the address of the base of the PCI core enumeration space. Should contain value 0x2, bit[13]=1				RESERVED 0	CardBus Mode	Force Clock Running ON	RESERVED 0								Function 0 Enable
3:2 [3:2]	Boot ROM size; 0x00 = 16KBytes, 0x01 = 32KBytes, 0x10 = 64KBytes, 0x11 = 128KBytes							Boot ROM Present (PXE)	Common Power PCI Power Management Specification v1.1 section 3.2.6							
5:4 [5:4]	SubSystemDeviceID (SSDID) also often called the SubSystemID; set by Vendor. For BCM NIC platforms this value is 0x970C															
7:6 [7:6]	SubSystemVendorID (SSVID); set by Vendor															
9:8 [9:8]	Device ID; set to 0x170C															
11:10 [B:A]	ClassCode [0:15]; set to 0x00 - see PCI v2.2 Specification section Appendix D															
13:12 [D:C]	D0 Power = 523 mW = 0x34							AuxCurrent Report = '1'	ClassCode [16:23]; set to 0x02 for Network Controller Class							
15:14 [F:E]	D3 Power = 329 mW = 0x20				SBBar0InitValue[11:8] = 0000b				D1 and D2 Power = 0 mW = 0000b				BAR0-IO	PME Source; set to 0x0		
17:16 [11:10]	SBAR0 Initial Value [12:15]; Set to 0x0				Ignore Clock Running request	Report PME from D3cold on VAUX	Report PME from D3cold Always	Report PME from D3hot	Report PME from D2	Report PME from D1	Report PME from D0	PME Enable	RESERVED 0			
19:18 [13:12]	SBAR0 Initial Value [16:31]; Set to 0x1800_															
21:20 [15:14]	CardBus CIS Pointer [0:15]															
23:22 [17:16]	CardBus CIS Pointer [16:31]															
39:24 [27:18]	RESERVED 0															
77:40 [4D:28]	DRIVER SPECIFIC AREA (FREE SPACE)															
83:78 [53:4E]	MAC Address (6 octets; 0x00-10-18-AA-BB-CC) where Broadcom OUI is 00-10-18															
91:90 [5B:5A]	ccLOCK (default 0x00)								Board Revision (0x01 as example)							
128:84 [80:54]	DRIVER SPECIFIC AREA (FREE SPACE)															
127:126 [7F:7E]	SPROM Checksum (CRC8)								SPROM Image Revision (0x01 as example)							

Note: If the PCI core detects that the first word (two-bytes) of the SPROM table are either all 0s or all 1s, it will set the SPROM values to their defaults.

Broadcom Corporation

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