Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!

Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832
Email & Skype: info@chipsmall.com Web: www.chipsmall.com
Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China

CoreTSE_AHB v2.1

Handbook

Microsemi

Table of Contents

Introduction

Core Overview

The CoreTSE_AHB provides 10/100/1000 Mbps Ethernet Media Access Controller (MAC) with a gigabit media independent interface (G/MII) or ten bit interface (TBI) to support 1000BASE-T and 1000BASE-X.

The CoreTSE_AHB has the following major interfaces:

- Advanced microcontroller bus architecture (AMBA®) high-performance bus (AHB)-master port interface allows data movement by direct memory access (DMA) engine between system memory and local Tx/Rx data buffers.
- G/MII or TBI physical layer (PHY) interface connects to Ethernet PHY.
- Management data input/output (MDIO) interface to communicate with the MDIO manageable device in the PHY.

The CoreTSE_AHB main functionality is provided by triple speed MAC core, which includes statistics gathering and station address functions. Statistics information is gathered from the data transmitted and received over the Ethernet link. Station address (SAL) feature provides address filtering capability.

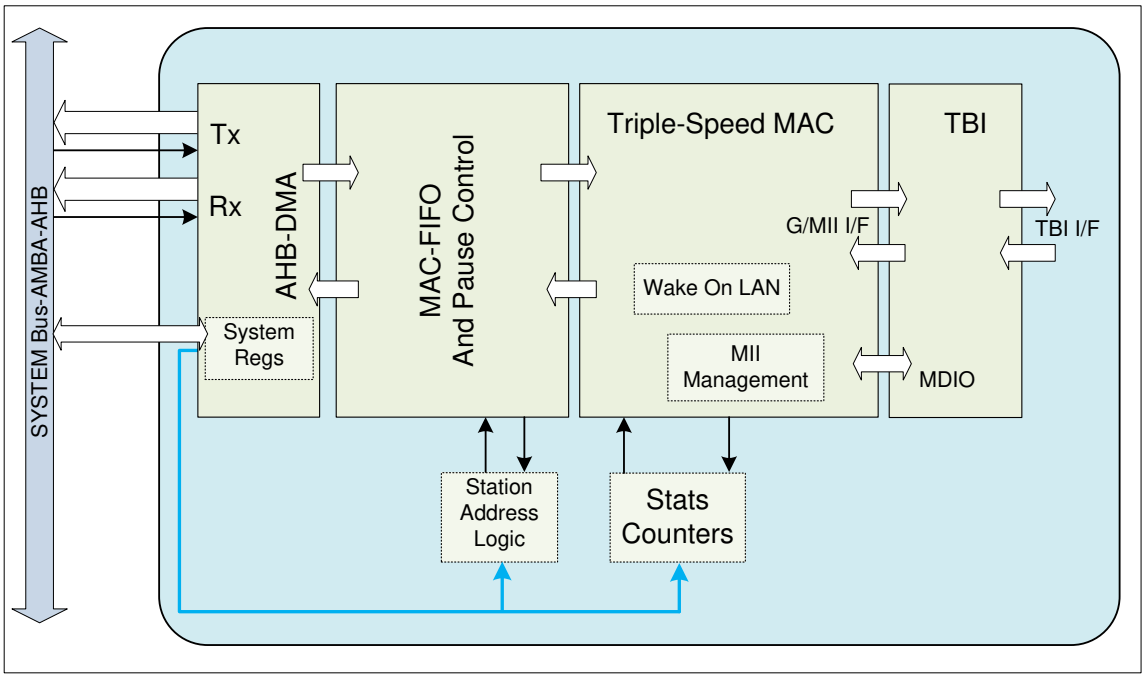

Figure 1 CoreTSE_AHB Block Diagram

Key Features

Following are the key features:

- Tri-Speed Ethernet MAC Core
- 10/100/1000 Mbps Operation
- Full-Duplex at 10/1000 Mbps
- Half-Duplex at 10/1000 Mbps
- Standard G/MII interface
- MDIO interface for PHY register access
- Ten-bit interface (TBI) for 1000Base-T or 1000Base-X support
- Wake on LAN (WoL) with Magic Packet Detection
- Frame Statistics Counters
- Destination Address Based Filtering

Utilization and Performance

Utilization and performance data is provided in Table 1, Table 2, Table 3, and Table 4 for the SmartFusion®2 and IGLOO®2 devices. The data is indicative only. In TBI mode (1000 Mbps), TXCLK, RXCLK, and GTXCLK performance was above 125 MHz.

Table 1 CoreTSE_AHB Device Utilization (G/MII, PACKET_SIZE = 256 Bytes, SAL-OFF, WAL-OFF, STATS- OFF)

Family	FPGA Resources			Utilization	
	4LUT	DFF	Total	Device	%
SmartFusion2	4,227	2769	6,996	M2S150T	4.78%
IGLOO2	4,227	2,769	6,996	M2GL150T	4.78%

Table 2 CoreTSE_AHB Device Utilization (G/MII, PACKET_SIZE = 32K Bytes, SAL-ON, WAL-ON, STATS-ON)

Family	FPGA Resources			Utilization	
	4LUT	DFF	Total	Device	%
SmartFusion2	9,300	5,945	15,245	M2S150T	10.4%
IGLOO2	9,300	5,945	15,245	M2GL150T	10.4%

Table 3 CoreTSE_AHB Device Utilization (TBI, PACKET_SIZE = 256 Bytes, SAL- OFF, WAL- OFF, STATS- OFF)

Family	FPGA Resources			Utilization	
	4LUT	DFF	Total	Device	%
SmartFusion2	6,140	3,787	9,927	M2S150T	6.79%
IGLOO2	6,140	3,787	9,927	M2GL150T	6.79%

Table 4 CoreTSE_AHB Device Utilization (TBI, PACKET_SIZE = 32K Bytes, SAL-ON, WAL-ON, STATS-ON)

Family	FPGA Resources			Utilization	
	4LUT	DFF	Total	Device	%
SmartFusion2	11,266	6,971	18,237	M2S150T	12.48%
IGLOO2	11,266	6,971	18,237	M2GL150T	12.48%

Functional Description

Triple Speed MAC

This core is a full-featured 10/100/1000 Mbps MAC with standard G/MII. The MAC has built in G/MII to TBI converter, which supports 1000 Mbps with TBI. The core is capable of full-duplex operation at 10, 100, or 1000 Mbps and of half duplex operation at 10 or 100 Mbps.

In half-duplex mode, the MAC adheres to the Carrier Sense Multiple Access/Collision Detect Access method as defined in IEEE 802.3 and its several supplements including IEEE 802.3u. In full-duplex mode, the MAC follows IEEE 802.3x, which ignores both carrier and collisions. Following each packet transmission or abortion, a transmit statistics vector is used for statistics collection.

The external PHY device presents packets to the MAC. The MAC scans the preamble searching for the start frame delimiter (SFD). When the SFD is found, the preamble and SFD are stripped and the frame is passed to the system. Following each frame reception, a Receive Statistics Vector is used for frame filtering and statistics collection.

CoreTSE_AHB supports PAUSE control frames. This core also includes optional support for Wake-On-Local-Area-Network module. The Wake on LAN (WoL) module detects both IEEE 802.3-compliant unicast frames with a destination address that matches the station address and packets that use AMD's Magic Packet™ Detection technology. The detection functionality can be enabled or disabled.

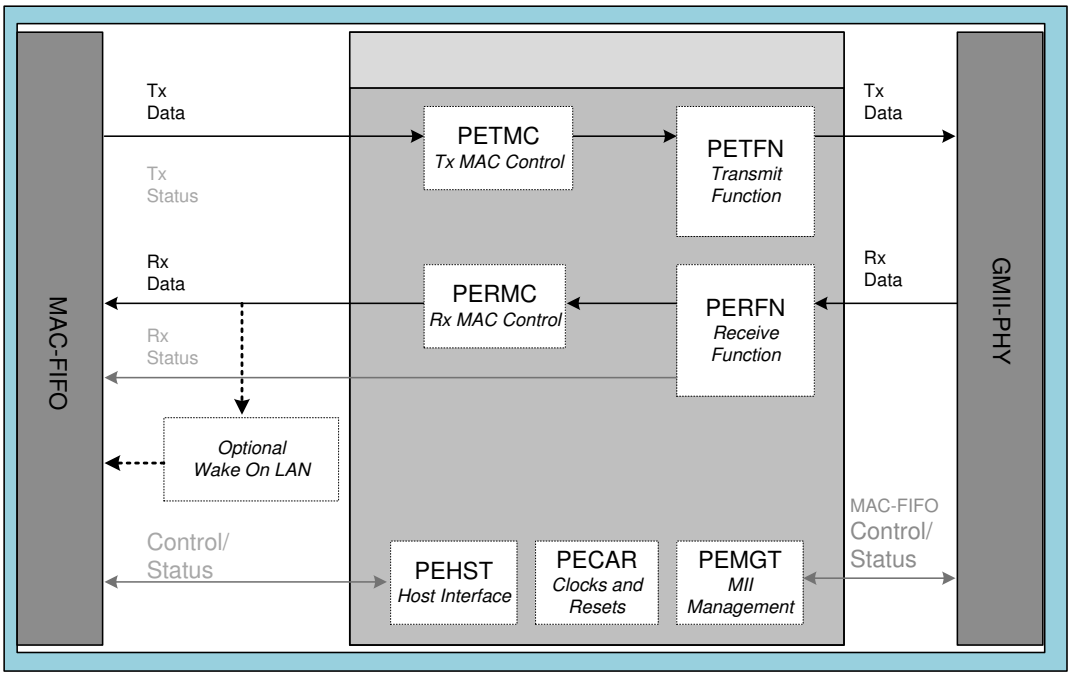

Figure 2 : Triple Speed MAC Functional Block Diagram

PAUSE Flow Control

MAC transmit logic (MACTL) provides native support for PAUSE flow control frames. PAUSE frames are control frames (frames with 0x8808 as the EtherType) with a particular DA (01-80-c2-00-00-01) and the opcode 0x0001. The FIFO-logic will automatically request to send a PAUSE frame by pulsing transmit-control-request (TCRQ) and providing the pause time value available on control-frame-register (CFPT [15:0]). Once a frame is received and detected as a control frame, MAC checks for the DA and the Opcode fields. If the DA is either the reserved multicast address used by PAUSE (01-80-c2-00-00-01) or the station's unique address, and the Opcode is 0x0001, then the Control frame is considered to be a PAUSE Control frame.

When a PAUSE Control frame is received:

- The MAC receive logic (MACRL) module indicates the MACTL to pause the stream of data frames and allows control frames transmission to the link partner. When either a PAUSE frame with a zero-value pause time is received or the MACRL pause timer expires, MACTL is considered to be unpaused and normal data frames gets resumed.
- The pause time value is loaded into the PERMC pause timer. This pause timer is a 16-bit down counter that decrements every pause quanta (a speed-independent constant of 64 byte-times). Whenever the pause time counter is non-zero, the MAC is considered to be paused and no data frames are sent.

Jumbo Frame Support

The CoreTSE_AHB supports jumbo frames that exceed the 1500 byte max of the standard Ethernet frame. When using jumbo frames the amount of idles that are present in the system will be reduced and therefore the frequency of clock compensation events will be lower. When supporting jumbo frames the clocking tolerance between the GTXCLK and the PMA_RXCLK0/1 is required to be 0ppm to account for the reduction in idles.

The Jumbo frame length transmitted / received by the CoreTSE_AHB is according to Maximum Frame Length (0x010) register configuration and supports up to 4000 bytes only.

Inter-Frame-Gap

MACRL provides the capability to filter frames that have less than a certain inter-frame-gap. The standard states that the inter-frame-gap should be 160 bit-times. This includes 96 bits of inter packet gap (IPG), 56 bits of preamble and 8 bits of start frame delimiter (SFD). To protect downstream logic from over-running, MACRL can be programmed with a minimum inter frame gap (IFG) parameter. The second of two back-to-back frames to violate the minimum IFG is dropped.

Address Detect

MACRL scans the frame and determine its address type. The 48-bit programmed station address is compared to each receive frame's DA. When they match, the unicast address detect (UCAD) is asserted. If the broadcast address is detected, MACRL asserts broadcast address detect (BCAD). If a multicast address is detected, the MAC asserts multicast address detect (MCAD).

Hash Table Support

MACRL supports hash table with up to 128 entries. Seven bits of the cyclic redundancy check (CRC) of the DA are used as the Hash Value (HASHV [6:0]).

Length Checking and Maximum Length Enforcement

MACRL can optionally compare the length field with the actual length of the data field portion of the frame. This is enabled through the MAC Configuration #2 register. MACRL first determines if the length/type field is a valid length. If so, it is compared with the data field length and any mismatches are updated to the receive statistics.

MACRL can limit the length of receive frames passed to the system. The maximum length is programmed through the Maximum Frame Length register. Frames which exceed this maximum are truncated.

Internal Loopback at G/MII

Asserting the internal loopback enable bit in MAC Configuration #1 register, enables MAC transmit output's looped back to the MAC receive inputs at G/MII interface.

Wake on Local Area Network (WoL)

The MAC -WoL is based on AMD's Magic Packet Detection technology.

The first step of the detection procedure is to scan the first twelve bytes of the frame, which contain Destination and Station addresses. Magic Packet detection is only carried out when the incoming frame's destination address matches the MAC's station address, or if the frame's destination address is a multicast or broadcast address.

After the first twelve bytes of the frame have matched, Core searches for the Magic Packet technology's defined preamble of six continuous aligned bytes with all bits asserted (0xFFh). Following a valid Magic Packet preamble, Core immediately expects 16 back-to-back repetitions of the six-byte MAC station address. Failure to achieve this exact pattern by a single byte at any time during the frame resets the circuitry back to the preamble search state.

After successful recognition of the Magic Packet payload or a successful compare of the MAC's station address with the incoming frame's destination address, the Interface Status register (bit field WakeOnLaneDetected) is asserted and status bit can only be cleared through assertion of the WakeOnLaneDetectedClear bit field of Interface Control register.

MDIO Management

Control and status is provided to and from the PHY through the two-wire MDIO management interface described in IEEE802.3u Clause22.

The MDIO write/read cycles are requested through the AHB slave. MAC performs a write cycle using the MDIO_PHYID, register address and 16-bit write data. MAC performs a read cycle using the MDIO_PHYID, register address and updates the Sixteen-bit read data into the MDIO Management Status register which can be read through AHB slave.

MAC FIFO

This core provides data queuing for increased throughput and sits between back-end, user-interface logic, and MAC core. The core provides clock-domain crossing, automatic pause frame handshaking, and graceful frame dropping.

The data is buffered between the system-interface and the MAC core by transmit and receive FIFOs. The FIFO size can be configured with PACKET_SIZE parameter.

In the design PACKET_SIZE is used as transmit FIFO address width, RAM's total frame data storage capability in bytes can be represented by the following equations:

TX RAM max frame data size = 2^ PACKET_SIZE * 4

RX RAM max frame data size = 2^ (PACKET_SIZE+1) * 4

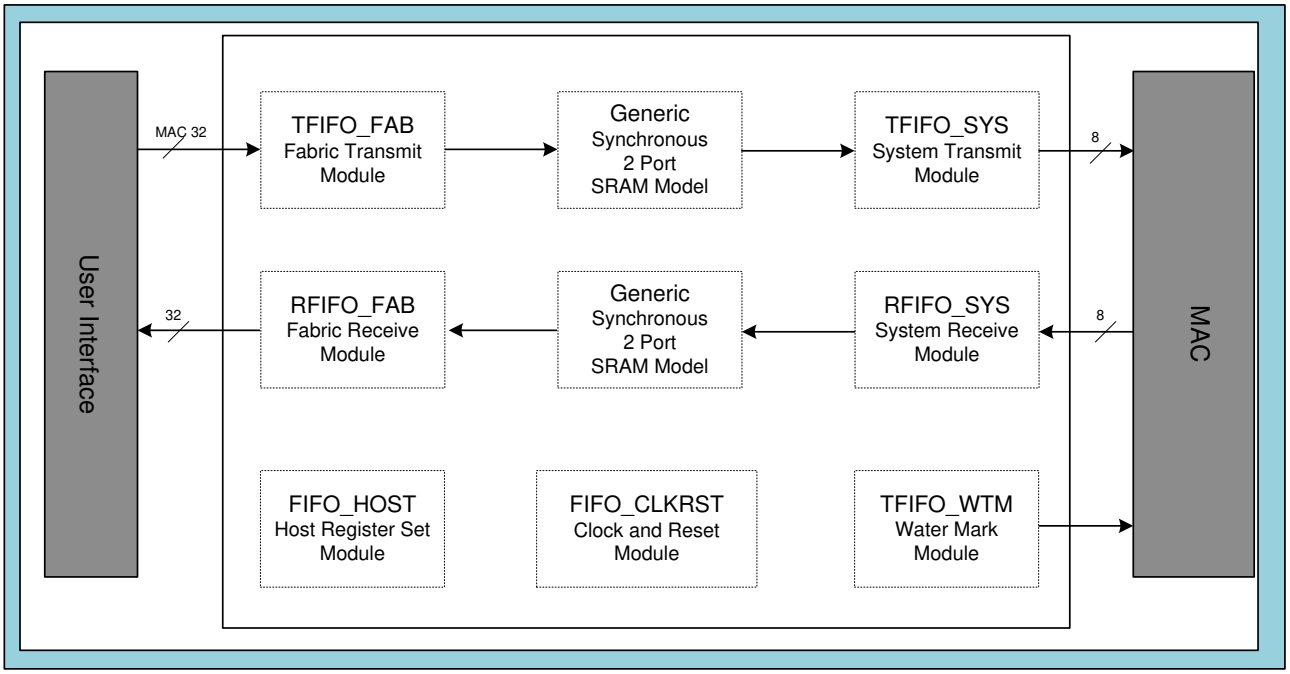

Figure 3 : MAC-FIFO Functional Block Diagram

Each RAM has additional associated control bits, which are additional to max frame data size.

Table 5 MAC-FIFO RAM Configurations

PACKET_SIZE (BYTES)	Transmit RAM Bit Dimensions (BITS)	Receive RAM Bit Dimensions (BITS)
256	64X39	NA
512	128X39	128X39
1K	256X39	256X39
2K	512X39	512X39
4K	1KX39	1KX39
8K	2KX39	2KX39
16K	4KX39	4KX39
32K	8KX39	8KX39
NA	NA	16KX39

AMBA-AHB Compliant DMA Engines

This module provides a DMA bridge between a host-system that uses an AMBA AHB™ bus and the Ethernet MAC and MAC-FIFO. It interfaces to the host-system through 32-bit AHB master and slave ports. On the MAC side of the module, it has a high-performance synchronous interface for DMA data transfer to and from the FIFO.

For ease of handling the software, transfers are handled using linked lists of transfer descriptors which together define one buffer in host memory for Tx operations and another for Rx operations. These buffers are typically configured as ring buffers, but this is up to the user to implement. Registers within the AHB-DMA provide control and status information regarding these transfers. These registers are accessed through the AHB slave port, alongside accesses to the AHB Slave interfaces on the MAC and the FIFO.

Figure 4 shows the AHB-DMA functional block diagram.

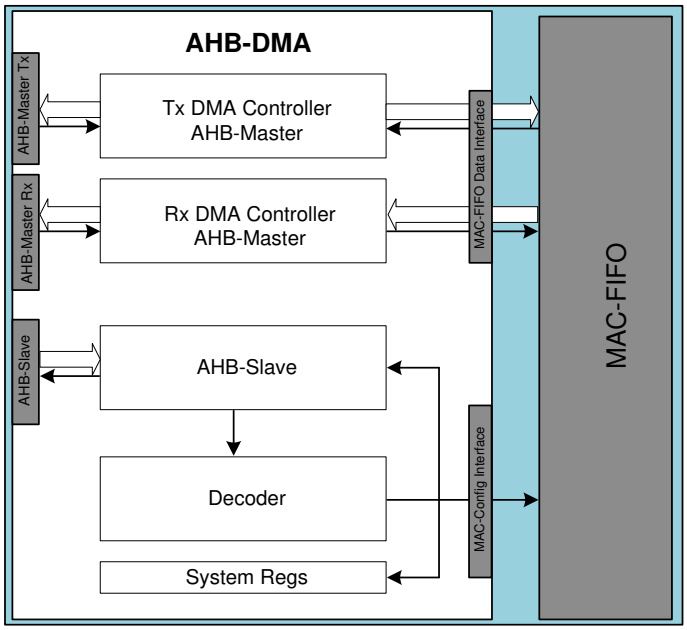

Figure 4 : AHB-DMA Functional Block Diagram

Interrupt Coalescing

The Scatter-Gather feature greatly improves DMA operational throughput by removing the system CPU from the real-time control of the DMA. However, this improved transfer throughput can overwhelm the CPU with transfer completion interrupts. This is solved by providing a programmable filter method called interrupt coalescing. The DMA interrupt coalescing feature is added beyond the traditional individual interrupt for each successful packet transfer. This can be enabled by setting corresponding Mask bit in the Interrupt Mask register. The default value of the interrupt threshold count is 1 and this must be updated as per the application requirements.

While DMA is operational, the Tx/Rx PktCnt is maintained. With each local interrupt on completion of event (TxPkt transmitted / RxPkt received), PktCnt is incremented. When the count reaches the coalescing threshold value coalescing interrupt bit is set in interrupt register. This interrupt remains asserted as long as the Tx/Rx-Count value in the DMA status register is non-zero. To acknowledge this coalescing interrupt, software must decrement PktCnt in the DMA Status register by writing 1 to the corresponding bit after acknowledging the event (TxPkt transmitted/RxPkt received).

Station Address Logic for Frame Filtering

This module provides a mechanism to statistically filter frames not intended for this node.

The MAC core performs DA comparison on all the received frames and provides three information signals: UCAD (Perfect DA match), MCAD, and BCAD along with seven most significant bits of the resulting CRC of DA. This information is used to perform a hashing algorithm, compare the result to a programmable hash table and then communicate to the FIFO logic to either delete or store the frame.

The programmability allows the user to assert any bits in a 128-bit hash table that corresponds to the desired Ethernet MAC DA. If the corresponding bit in the table is set, the frame will be accepted. In addition, hashing can selectively be performed on unicast addresses or multicast addresses.

Statistics Counters Logic

This module has separate counters, which simply counts or accumulate conditions that occur upon packets are transmitted and received. These counters support remote network monitoring (RMON) management information base (MIB) group 1, RMON MIB group 2, RMON MIB group 3, RMON MIB group 9, RMON MIB 2, and the dot 3 Ethernet MIB.

COMMA Alignment Logic

The PHY layer includes COMMA alignment logic in the receive path. This logic detects COMMA data and aligns the 10-bit data to the proper word boundary before passing the data to the receive path.

Ten bit interface

This module takes the transmit G/MII data stream, encodes it into 10-bit symbols and presents 10-bit interface data to SERDES. Packet data replication is used to match data rates for the different modes of the MII to the transmit clock. In the receive direction de-serialized 10-bit symbols are decoded and converted into the receive G/MII signal set. Packet data under sampling is used to match data rates for the different modes of the MII to the TBI receive clock.

The design uses transmit, receive, and synchronization state machines as specified in Clause 36 of IEEE 802.3z. Also included auto-negotiation (AN) for 1000BASE-X, which is used to exchange information between the link partners. This module is managed and monitored through the MDIO management interface. The extended set of management registers is provided.

Both the transmit and receive paths leverage the physical coding sub layer and the Auto-negotiation sub-layers of the IEEE 802.3z specification, as contained in Clauses 36 and 37. For complete clock domain isolation of the TBI from the MAC, both transmit and receive elasticity FIFOs are used.

The control information exchanged differs from the IEEE specification. Instead of using the ability advertisement, the PHY sends the control information through its Tx_config_Reg [15:0], as listed in Table 6. Upon receiving control information, the MAC acknowledges the update of the control information by asserting bit 14 of its Tx_config_Reg [15:0].

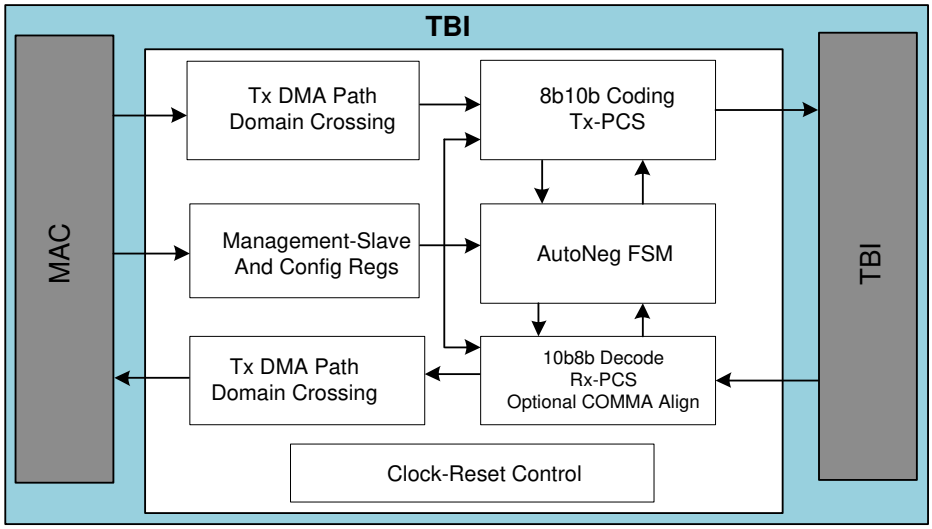

Figure 5 : TBI Functional Block Diagram

In order to maintain a constant clock frequency at the PHY interface for all MAC speeds, the MII bus data must be replicated internally to the TBI. Nibble packet data transmitted by a 100 Mbps MAC must be aligned, concatenated, and replicated 10 times. Nibble packet data transmitted by a 10 Mbps MAC must be aligned, concatenated, and replicated 100 times.

Table 6 TBI Auto-Negotiation Control Information Sent/Received

Bit	Tx_config from PHY to MAC	Tx_config from MAC to PHY
15	Link: 1: Link up 0: link down	0: Reserved
14	Reserved for AN ACK.	1
13	0: Reserved	0: Reserved
12	Duplex mode: 1: Full 0: Half	0: Reserved

Bit	Tx_config from PHY to MAC	Tx_config from MAC to PHY
11:10	Speed: Bit 11, 10: 11: Reserved 10: 1000 Mbps 00:10 Mbps	0: Reserved
9:1	0: Reserved	0: Reserved
0	1	1

Packet data received by the TBI through the PHY must be under sampled by a factor of 10 before being sent to a 100 Mbps MAC. Packet data received by the TBI through the PHY must be under sampled by a factor of 100 before being sent to a 10 Mbps MAC. For half-duplex functionality, carrier sense is inferred from RXDV, and collision is derived from the simultaneous assertion of TXEN and RXDV.

Programmer Guide

This section has all the information regarding usage of the core.

Functional Overview

The AHB-DMA Bridge core includes two-channel DMA-Controller for transmit and receive data path operations. The transfer of data in either direction typically uses a ring buffer defined within host memory. The ring buffers for transmit and receive operations are defined by a closed linked list of Tx/Rx descriptors. The two ring buffers are formed of equal-sized segments, each of which is 32-bit aligned and is capable of storing a packet of up to the maximum size of packet transferred. The way transmit operations and receive operations are carried is as described below. The requirement of the AHB specification that burst transfers must not cross 1Kbyte boundaries is handled seamlessly by the AHB-DMA Bridge core and does not affect the operation of the Ethernet components of the system. Once initialized, the DMA controller can be left unattended to continuously fill/empty the specified ring buffers. Software can either use the DMA interrupts generated or poll semaphore bits within the descriptors to maintain synchronization with the packet streams.

Descriptor Information

Before any DMA transfers can be carried out, two sets of descriptors need to be initialized in the host memory, one for transmit operations and the other for receive operations. Each set of descriptors takes the form of a linked list (typically closed to form a ring buffer).

Table 7 DMATx/Rx Descriptor Information

ADDRESS	REGISTER	FUNCTION	SIZE
+0h	PacketStartAddr	Start address for the packet data	32 bits
+4h	PacketSize	Size of packet, Overrides and Empty Flag	32 bits
+8h	NextDescriptor	Location of next descriptor	32 bits

The entry point into the buffer used at the start of any sequence of transfers is given by the Descriptor picked out by the DMATx/RxDescriptor register. Each descriptor comprises a sequence of three 32-bit memory locations as listed in Table 8.

Table 8 AHB-DMA Descriptor Information

Addr Offset	Function
0x000	Start address for packet data **[31:2] (R/W) Top 30 bits of Packet Start Address. Default 0x00** The built-in DMA controller reads this register to discover the location in host memory of the first byte of data. **[1:0] (R/W) Ignored. Default is 0.** Ignored by the DMA controller, since it is a requirement of the system that all transfers are 32-bit aligned in the host memory.
0x004	Packet Information **[31] (R/W) Empty Flag** For transmit operations, this bit indicates the availability of the data associated with the packet. For receive operations, this bit indicates the availability of the specified location to store the received packet. The setting of this flag is used to validate the descriptor. *Note: On successful completion of a transmit operation, the DMA controller writes 1 to this location to indicate that the associated data has been transferred from this location. On successful completion of a receive operation, the DMA controller writes 0 to this location to indicate that this location has been used to store the received packet. The first action ensures that the data cannot be accidentally transferred twice, the second action ensures that received data is not accidentally overwritten by a subsequent packet.*

Addr Offset	Function
	[20:16] (R/W) FTPP Overrides 5-bit field containing the per-packet override flags signaled to the FIFO during packet transmission. The bits are encoded as follows: 20: FIFO Transmit Control Frame [19:18]: FIFO Transmit Per-Packet PAD Mode 17: FIFO Transmit Per-Packet Generate FCS 16: FIFO Transmit Per-Packet Enable **[15:0] (R/W) Packet Size** 16-bit field which, for transmit operations, gives the size of packet to be transferred in bytes. In receive operations, the DMA controller writes the number of bytes received to this field: the value of this field prior to the transfer being made is ignored.
0x008	Pointer to Next Descriptor **[31:2] (R/W) Top 30 bits of Descriptor Address** The built-in DMA controller reads this register to discover the location in host memory of the descriptor for the next packet in the sequence. The descriptors should form a closed linked list. **[1:0] (R/W) Ignored. Default 0x0** Ignored by the DMA controller, since it is a requirement of the system that all descriptors are 32-bit aligned in host memory

Transmit Operation

Before any packets can be transmitted, a group of Tx descriptors needs to be set up to define the ring buffer used for transmit operations. The start addresses set for the different segments of the ring buffer are required to be 32-bit aligned and should be spaced to give segments of equal size, each able to handle a packet of the maximum size to be transferred. In addition, the PacketSize components of these descriptors should initially be written to have 1 in bit 31 (the Empty Flag) to indicate that the ring buffer does not currently contain any valid data.

The bottom four bits of the DMAIntrMask register also need to be set to specify which Tx DMA events will cause a DMA interrupt to be generated. The data for one or more transmit packets should then be placed in contiguous segments of the ring buffer and the PacketSize component of the descriptor associated with these segments amended both to record the size of the packet placed in the buffer and to set the Empty Flag to 0 to indicate the presence of valid data. (Further packets for transmission can be written to the ring buffer as required, as long as segments are available within the ring buffer to accommodate these packets. Available segments are indicated by a 1 in bit 31 of the PacketSize component of the associated Tx descriptor.)

The location of the descriptor corresponding to the required entry point into the Tx ring buffer should then be written to the DMATxDescriptor register and the TxEnable bit (bit 0 of the DMATxCtrl register) set to enable DMA transfer of transmit packets.

The built-in DMA controller then reads DMATxDescriptor to discover the location of the first Tx descriptor, then read that descriptor initially to check the validity of the associated packet (indicated by the Empty Flag in bit 31 of the PacketSize component of the descriptor being set to 0) then to discover the start address of the packet to be transmitted and its size. (If the Empty Flag is 1, the descriptor is not currently associated with valid data. Where this is the case, the DMA controller terminates the sequence of transmit packet transfers, set the TxUnderrun bit in the DMATxStatus register and clear the TxEnable bit in the DMATxCtrl register. If enabled, an interrupt is generated with the DMAInterrupts register showing TxUnderrun as the source of this interrupt. Any further transfers require the DMATxDescriptor register to be updated to record the start position in the ring buffer that is now required and to set the TxEnable bit to 1 again.)

Once the transfer is completed successfully, the DMA controller writes 1 to bit-31 of the PacketSize component of the descriptor. The TxPktSent flag in the DMATxStatus register is also set (if not already set), the TxPktSent interrupt is generated (if enabled) and the TxPktCount recorded in bits [23:16] of the DMATxStatus register is incremented by 1.

The DMA controller then moves on to process any packet stored in the next segment of the ring buffer. The location of the descriptor associated with the next segment in the ring will have been already read from the NextDescriptor component of the current descriptor.

If a bus error occurs, the DMA controller terminates the sequence of transmit packet transfers, set the Bus Error bit in the DMATxStatus register and clear the TxEnable bit in the DMATxCtrl register. If enabled, an interrupt is generated with the DMAInterrupts register showing a Tx Bus Error as the source of this interrupt. Any further transfers require the DMATxDescriptor register to be updated to record the new start position in the ring buffer and the TxEnable bit to be set to 1 again.

Receive Operation

Before any packets can be received, a group of Rx descriptors needs to be set up to define the ring buffer used for receive operations. The start addresses set for the different segments of the ring buffer are required to be 32-bit aligned and should be spaced to give segments of equal size, each able to handle a packet of the maximum size to be transferred. In addition, the PacketSize components of these descriptors should initially be written to have 1 in bit 31 (the Empty Flag) to indicate that the ring buffer does not currently contain any received packets. Bits[7:4] of the DMAIntrMask register also need to be set to specify which Rx DMA events cause a DMA interrupt to be generated. With these items in place, the location of the descriptor corresponding to the required entry point into the Rx ring buffer should be written to the DMARxDescriptor register and the RxEnable bit (bit 0 of the DMARxCtrl register) set to enable DMA transfer of receive packets.

The built-in DMA controller then reads DMARxDescriptor to discover the location of the first Rx descriptor, then reads that descriptor initially to check that the associated area of host memory is available for storing the received packet (indicated by the Empty Flag in bit 31 of the PacketSize component of the descriptor being set to 1) then to discover the start address of this storage area.

If the Empty Flag is 0, this suggests that this storage area already contains a packet that has not yet been read by the host software. Where this is the case, the DMA controller will terminate the sequence of Receive packet transfers, set the RxOverflow bit in the DMARxStatus register, and clear the RxEnable bit in the DMARxCtrl register. If enabled, an interrupt will be generated with the DMAInterrupts register showing RxOverflow as the source of this interrupt. Any further transfers require the DMARxDescriptor register to be updated to record the start position in the ring buffer that is now required and the RxEnable bit to be set to 1 again.

If the transfer is completed successfully, the DMA controller records the number of bytes transferred in bits[11:0] of the PacketSize component of the descriptor and write 0 in bit 31 to record that a packet has been stored in the ring buffer. The RxPktReceived flag in the DMARxStatus register will also be set (if not already set), a RxPktReceived interrupt will be generated (if enabled) and the RxPktCount recorded in bits[23:16] of that register incremented by 1. If Rx FIFO ready continues to be asserted, the DMA controller will then move on to transfer the next packet in the next segment of the ring buffer. The location of the descriptor associated with the next segment in the ring will have been already read from the NextDescriptor component of the current descriptor.

Software should respond to the RxPktReceived interrupt by reading the packet from its location in the ring buffer and then setting the Empty Flag in the descriptor to 1 again to mark this segment of the ring buffer as available for storing further received packets.

If a bus error occurs, the DMA controller will terminate the sequence of Receive packet transfers, set the Bus Error bit in the DMARxStatus register and clear the RxEnable bit in the DMARxCtrl register. If enabled, an interrupt will be generated with the DMAInterrupts register showing an Rx Bus Error as the source of this interrupt. Any further transfers will require the DMARxDescriptor register to be updated to record the start position in the ring buffer that is now required and the RxEnable bit to be set to 1 again.

Register Map

The external AHB master uses a 32-bit AHB slave interface for accessing control and status registers.

Table 9 Core Register MAP

Address Offset	Function
0x000 – 0x044	Access to MAC core registers
0x048 – 0x07C	Access to FIFO core registers
0x080 – 0x17F	Access to Statistics Counters core registers 0x080 – 0x13C are valid addresses
0x180 – 0x1BF	Access to AHB-DMA registers AHB: 0x180 – 0x19C are valid addresses
0x1C0 – 0x1FF	Access to System Registers (SAL and miscellaneous controls) 0x1C0 – 0x1D4 are valid addresses

MAC Core Registers

Table 10 Control/Status Registers

Address[9:0]	Function
0x000	MAC Configuration #1 **[31] (R/W) SOFT RESET: Default 1** Setting this bit puts all modules within the MAC in reset except the AHB slave interface. **[30] (R/W) SIMULATION RESET: Default 0** Setting this bit resets those registers, such as the random backoff timer, which are not controlled by normal resets. (simulation only) **[29:20] Reserved** **[19] (R/W) RESET RX MAC CONTROL: Default 0** Setting this bit puts the PERMC Receive MAC Control block in reset. This block detects Control frames and contains the pause timers. **[18] (R/W) RESET TX MAC CONTROL: Default 0** Setting this bit puts the PETMC Transmit MAC Control block in reset. This block multiplexes data and Control frame transfers. It also responds to XOFF(transmit OFF) PAUSE Control frames. **[17] (R/W) RESET RX FUNCTION: Default 0** Setting this bit puts the PERFN Receive Function block in reset. This block performs the receive frame protocol. **[16] (R/W) RESET TX FUNCTION: Default 0** Setting this bit puts the PETFN Transmit Function block in reset. This block performs the frame transmission protocol. **[15:9] Reserved** **[8] (R/W) LOOP BACK: Default 0** Setting this bit causes the PETFN MAC Transmit outputs to be looped back to the MAC Receive inputs. Clearing this bit results in normal operation. **[7:6] Reserved** **[5] (R/W) RECEIVE FLOW CONTROL ENABLE: Default 0** Setting this bit causes the PERFN Receive MAC Control to detect and act on PAUSE Flow Control frames. Clearing this bit causes the Receive MAC Control to ignore PAUSE Flow Control frames. **[4] (R/W) TRANSMIT FLOW CONTROL ENABLE: Default 0** Setting this bit allows the PETMC Transmit MAC Control to send PAUSE Flow Control frames when requested by the system. Clearing this bit prevents the Transmit MAC Control from sending Flow Control frames. **[3] (RO) SYNCHRONIZED RECEIVE ENABLE:** Receive Enable synchronized to the receive stream. **[2] (R/W) RECEIVE ENABLE: Default 0** Setting this bit allows the MAC to receive frames from the PHY. Clearing this bit prevents the reception of frames. **[1] (RO) SYNCHRONIZED TRANSMIT ENABLE:** Transmit Enable synchronized to the transmit stream. **[0] (R/W) TRANSMIT ENABLE: Default 0** Setting this bit allows the MAC to transmit frames from the system. Clearing this bit will prevent the transmission of frames.

Address[9:0]	Function
0x004	MAC Configuration #2 **[31:16] Reserved** **[15:12] (R/W) PREAMBLE LENGTH: Default 0x7** This field determines the length of the preamble field of the packet, in bytes. **[11:10] Reserved** **[9:8] (R/W) INTERFACE MODE: Default 0x10** This field determines the type of MAC interface in G/MII mode, for TBI the interface mode should be 0x10 2'b00: MAC Tx/Rx represents MII 10Mbps interface (Nibble Mode) 2'b01: MAC Tx/Rx represents MII 100Mbps interface (Nibble Mode) 2'b10: MAC Tx/Rx represents GMII 1000Mbps interface (Byte Mode) 2'b11: Reserved **[7:6] Reserved** **[5] (R/W) HUGE FRAME ENABLE: Default 0** Setting this bit allows frames longer than the MAXIMUM FRAME LENGTH to be transmitted and received. Clear this bit to have the MAC limit the length of frames at the MAXIMUM FRAME LENGTH value. (Maximum Frame Length is set in separate Maximum Frame Length register.) **[4] (R/W) LENGTH FIELD CHECKING: Default 0** Setting this bit causes the MAC to check the frame's length field to ensure it matches the actual data field length. Clear this bit if no length field checking is desired. **[3] Reserved** **[2] (R/W) PAD / CRC ENABLE: Default 0** Set this bit to have the MAC pad all short frames and append a CRC to every frame whether or not padding was required. Clear this bit if frames presented to the MAC have a valid length and contain a CRC. **[1] (R/W) CRC ENABLE: Default 0** Set this bit to have the MAC append a CRC to all frames. Clear this bit if frames presented to the MAC have a valid length and contain a valid CRC. If the PAD/CRC ENABLE configuration bit or the per-packet PAD/CRC ENABLE is set, CRC ENABLE is ignored. **[0] (R/W) FULL-DUPLEX: Default 0** Setting this bit configures the MAC to operate in full-duplex mode. Clearing this bit configures the MAC to operate in half-duplex mode only.
0x008	IPG / IFG **[31] Reserved** **[30:24] (R/W) NON-BACK-TO-BACK INTER-PACKET-GAP PART1 (IPGR1):** This programmable field represents the optional carrierSense window referenced in IEEE 802.3/4.2.3.2.1 Carrier Deference. If a carrier is detected during the timing of IPGR1, the MAC defers to the carrier. If, however, the carrier becomes active after IPGR1, the MAC continues timing IPGR2 and transmits, knowingly causing a collision. This ensures fair access to the medium. The permitted range of values is 0x0 to IPGR2. **Default is 0x40** (64d) which follows the two-thirds/one-thirds guideline. **[23] Reserved** **[22:16] (R/W) NON-BACK-TO-BACK INTER-PACKET-GAP PART2 (IPGR2):** This programmable field represents the Non-Back-to-Back Inter-Packet-Gap in bit times. Default is 0x60 (96d), which represents the minimum IPG of 96 bits.

Address[9:0]	Function
	[15:8] (R/W) MINIMUM IFG ENFORCEMENT: Default 0x50 This programmable field represents the minimum size of IFG to enforce between frames (expressed in bit times). A frame whose IFG is less than that programmed is dropped. The default setting of 0x50 (80d) represents half of the nominal minimum IFG which is 160 bits. **[7] Reserved** **[6:0] (R/W) BACK-TO-BACK INTER-PACKET-GAP: Default 0x60** This programmable field represents the IPG between Back-to-Back packets (expressed in bit times). This is the IPG parameter used exclusively in full-duplex mode when two transmit packets are sent back-to-back. Set this field to the desired number of bits. The default setting of 0x60 (96d) represents the minimum IPG of 96 bits.
0x00C	Half-Duplex **[31:24] Reserved** **[23:20] (R/W) ALTERNATE BINARY EXPONENTIAL BACKOFF TRUNCATION: Default 0xA** This field is used when ALTERNATE BINARY EXPONENTIAL BACKOFF ENABLE is set. The value programmed is substituted for the Ethernet standard value of ten. **[19] (R/W) ALTERNATE BINARY EXPONENTIAL BACKOFF ENABLE: Default 0** Setting this bit configures the Tx MAC to use the ALTERNATE BINARY EXPONENTIAL BACKOFF TRUNCATION setting instead of the 802.3 standard tenth collisions. The Standard specifies that any collision after the tenth uses 2^{10}-1 as the maximum backoff time. Clearing this bit causes the Tx MAC to follow the standard binary exponential backoff rule. **[18] (R/W) BACKPRESSURE NO BACKOFF: Default 0** Setting this bit configures the Tx MAC to immediately re-transmit following a collision during backpressure operation. Clearing this bit causes the Tx MAC to follow the binary exponential backoff rule. **[17] (R/W) NO BACKOFF: Default 0** Setting this bit configures the Tx MAC to immediately re-transmit following a collision. Clearing this bit causes the Tx MAC to follow the binary exponential backoff rule. **[16] (R/W) EXCESSIVE DEFER: Default 1** Setting this bit configures the Tx MAC to allow the transmission of a packet that has been excessively deferred. Clearing this bit causes the Tx MAC to abort the transmission of a packet that has been excessively deferred. **[15:12] (R/W) RETRANSMISSION MAXIMUM: Default 0xF** This is a programmable field specifying the number of retransmission attempts following a collision before aborting the packet due to excessive collisions. The Standard specifies the maximum number of attempts to be 0xF (15d). **[11:10] Reserved** **[9:0] (R/W) COLLISION WINDOW: Default 0x37** This programmable field represents the slot time or collision window during which collisions might occur in a properly configured network. Since the collision window starts at the beginning of transmission, the preamble and SFD are included. The default of 0x37 (55d) corresponds to the count of frame bytes at the end of the window.

Address[9:0]	Function
0x010	Maximum Frame Length **[31:16] Reserved** **[15:0] (R/W) MAXIMUM FRAME LENGTH: Default 0x07D0 (2000 d)** This programmable field sets the maximum frame size in both the transmit and receive directions.
0x014	Control Frame extended parameter (Used for pause frame) **[31:16] Reserved** **[15:0] (R/W) CFEP: Default 0x0000**
0x018	Control Frame parameter (Used for pause Value) **[31:16] Reserved** **[15:0] (R/W) CFPT: Default 0xFFFF**
0x01C	Test Register **[31:4] Reserved** **[3] (R/W) MAXIMUM BACKOFF: Default 0** Setting this bit causes the MAC to backoff for the maximum possible length of time. This test bit is used to predict backoff times in Half-Duplex mode. **[2] (R/W) REGISTERED TRANSMIT FLOW ENABLE: Default 0** Registered Transmit half-duplex Flow Enable. **[1] (R/W) TEST PAUSE: Default 0** Setting this bit allows the MAC to be paused through the AHB slave interface for testing purposes. **[0] (R/W) SHORTCUT SLOT TIME: Default 0** This bit allows the slot time counter to expire regardless of the current count. This bit is for testing purposes only.
0x020	MDIO Mgmt: Configuration **[31] (R/W) RESET MDIO MGMT: Default 0** Setting this bit resets MDIO Mgmt (provided by the PEMGT module). Clearing this bit allows MDIO Mgmt to perform Mgmt read/write cycles as requested via the AHB Slave interface. **[30:6] Reserved** **[5] (R/W) SCAN AUTO INCREMENT: Default 0** Setting this bit causes MDIO Mgmt to continually read from a set of PHYs of contiguous address space. The starting address of the PHY is specified by the content of the PHY address field recorded in the MDIO Mgmt Address register. The next PHY to be read will be PHY address + 1. The last PHY to be queried in this read sequence will be the one residing at address 0x31, after which the read sequence returns to the PHY specified by the PHY address field. **[4] (R/W) PREAMBLE SUPPRESSION: Default 0** Setting this bit causes MDIO Mgmt to suppress preamble generation and reduce the Mgmt cycle from 64 clocks to 32 clocks. This is in accordance with IEEE 802.3/22.2.4.4.2. Clearing this bit causes MDIO Mgmt to perform Mgmt read/write cycles with the 64 clocks of preamble. **[2:0] (R/W) MGMT CLOCK SELECT: Default 0x0** This field determines the clock frequency of the Mgmt Clock (MDC). Below – MGMT Clock Select Encoding to determine how to program this field. HCLK is the source clock. 3’b000: Source clock divided by 4 3’b001: Source clock divided by 4

Address[9:0]	Function
	3'b010: Source clock divided by 6 3'b011: Source clock divided by 8 3'b100: Source clock divided by 10 3'b101: Source clock divided by 14 3'b110: Source clock divided by 20 3'b111: Source clock divided by 28
0x024	MDIO Mgmt: Command **[31:2] Reserved** **[1] (R/W) SCAN CYCLE: Default 0** This bit causes MDIO Mgmt to perform Read cycles continuously. This is useful for monitoring Link Fail for example. **[0] (R/W) READ CYCLE: Default 0** This bit causes MDIO Mgmt to perform a single Read cycle. The Read data is returned in MDIO Mgmt Status Register.
0x028	MDIO Mgmt: Address **[31:13] Reserved** **[12:8] (R/W) PHY ADDRESS: Default 0x0** This field represents the 5-bit PHY Address field used in Mgmt cycles. Up to 31 PHYs can be addressed. **[7:5] Reserved** **[4:0] (R/W) REGISTER ADDRESS: Default 0x0** This field represents the 5-bit Register Address field of Mgmt cycles.
0x02C	MDIO Mgmt: Control **[31:16] Reserved** **[15:0] (WO) MDIO MGMT CONTROL (PHY Control):Default 0x0** When written, an MDIO Mgmt write cycle is performed using the 16-bit data and the pre-configured PHY and Register addresses from the MDIO Mgmt Address Register.
0x030	MDIO Mgmt: Status **[31:16] Reserved** **[15:0] (RO) MDIO MGMT STATUS (PHY STATUS):** Following an MDIO Mgmt Read Cycle, the 16-bit data can be read from this location
0x034	MDIO Mgmt: Indicators **[31:3] Reserved** **[2] (RO) NOT VALID: Default 0** When 1 is returned - indicates MDIO Mgmt Read cycle has not completed and the Read Data is not yet valid. **[1] (RO) SCANNING: Default 0** When 1 is returned - indicates a scan operation (continuous MDIO Mgmt Read cycles) is in progress. **[0] (RO) BUSY: Default 0** When 1 is returned - indicates MDIO Mgmt block is currently performing an MDIO Mgmt Read or Write cycle.

Address[9:0]	Function
0x038	Interface Control **[31] (R/W) RESET INTERFACE MODULE: Default 0** Setting this bit resets the Interface module. Clearing this bit allows for normal operation. **[30:6] Reserved** **[7] (W/R) WoL: Unicast match enable: Default 0** Setting this bit configures WoL module to enable WakeOnLaneDetected assertion based on Unicast match **[6] (W/R) WoL: Magic Packet detection enable: Default 0** Setting this bit configures WoL module to enable WakeOnLaneDetected assertion based on magic packet detection **[5] (W/R) WoL: WakeOnLaneDetectedClear status clear: Default 0** When this bit is asserted, WakeOnLaneDetected status is held low. When this bit is cleared, WakeOnLaneDetected may become asserted appropriately. **[4] (W/R) Stats Counters – Auto clear counters on read: Default 0** Setting this bit enables auto-clear-on-read feature for all the counters **[3] (W/R) Stats Counters – Clear All counters: Default 0** Setting this bit clears all the statistics counters. **[2] (W/R) Stats Counters – Module enable: Default 0** Setting this bit enables statistics counter module. **[1:0] Reserved**
0x03C	Interface Status **[30:11] Reserved** **[10] (RO/LH) WakeOnLaneDetected:** This bit is only used when the optional WoL module is integrated It is set when the MAC detects a Magic Packet and stays high until it is cleared by the assertion of WakeOnLaneDetectedClear. Its reset value is low. **[9] (RO/LH) EXCESS DEFER:** This bit sets when the MAC excessively defers a transmission. It clears when read. This bit latches high. Excessive Deferred is a condition when the MAC has deferred sending a packet for a time longer than the length of two maximum length frames or 3036 bytes time. **[8:4] Reserved** **[3] (RO) LINK FAIL:** When read as a 1, the MDIO management module has read the PHY link fail register to be 1. When read as a 0, the MDIO management module has read the PHY link fail register to be 0. Note that for asynchronous host accesses, this bit must be read at least once every scan read cycle of the PHY. **[2:0] Reserved**
0x040	Station Address Lower Register - Default 0x0000_0000 [31:24] (W/R) First octet of the DA in the frame [23:16] (W/R) Second octet of the DA in the frame [15: 8] (W/R) Third octet of the DA in the frame [7: 0] (W/R) Fourth octet of the DA in the frame
0x044	Station Address Higher Register Default 0x0000_0000 [31:24] (W/R) Fifth octet of the DA in the frame [23:16] (W/R) Sixth octet of the DA in the frame [15:0] Reserved

MAC-FIFO Core Registers

Table 11 MAC-FIFO Core Registers

Address[9:0]	Function
0x048	MAC-FIFO Configuration Register 0 **[31:21] Reserved** **[20] (RO) ftfenrply: Default 0** When asserted, the TFIFO_FAB module is enabled. When negated, the TFIFO_FAB module is disabled. The bit should be polled until it reaches the expected value. **[19] (RO) stfenrply: Default 0** When asserted, the TFIFO_SYS module is enabled. When negated, the TFIFO_SYS module is disabled. The bit should be polled until it reaches the expected value. **[18] (RO) frfenrply: Default 0** When asserted, the RFIFO_FAB module is enabled. When negated, the RFIFO_FAB module is disabled. The bit should be polled until it reaches the expected value. **[17] (RO) srfenrply: Default 0** When asserted, the RFIFO_SYS module is enabled. When negated, the RFIFO_SYS module is disabled. The bit should be polled until it reaches the expected value. **[16] (RO) wtmenrply: Default 0** When asserted, the TFIFO_WTM module is enabled. When negated, the TFIFO_WTM module is disabled. The bit should be polled until it reaches the expected value. **[15:13] Reserved** **[12] (R/W) ftfenreq: Default 0** When asserted, requests enabling of the TFIFO_FAB module. When negated, requests disabling of the TFIFO_FAB module. **[11] (R/W) stfenreq: Default 0** When asserted, requests enabling of the TFIFO_SYS module. When negated, requests disabling of the TFIFO_SYS module. **[10] (R/W) frfenreq: Default 0** When asserted, requests enabling of the RFIFO_FAB module. When negated, requests disabling of the RFIFO_FAB module. **[9] (R/W) srfenreq: Default 0** When asserted, requests enabling of the RFIFO_SYS module. When negated, requests disabling of the RFIFO_SYS module. **[8] (R/W) wtmenreq: Default 0** When asserted, requests enabling of the TFIFO_WTM module. When negated, requests disabling of the TFIFO_WTM module. **[7:5] Reserved** **[4] (R/W) hstrstft: Default 1** When asserted this bit will place the TFIFO_FAB module in reset. **[3] (R/W) hstrstst: Default 1** When asserted this bit will place the TFIFO_SYS module in reset.

Address[9:0]	Function
	[2] (R/W) hstrstfr: Default 1 When asserted this bit will place the RFIFO_FAB module in reset. **[1] (R/W) hstrstsr: Default 1** When asserted this bit will place the RFIFO_SYS module in reset. **[0] (R/W) hstrstwt: Default 1** When asserted this bit will place the TFIFO_WTM module in reset.
0x04C	MAC-FIFO Configuration Register 1 **[31:28] Reserved** **[27:16] (R/W) cfgsrth[11:0]: Default 0xFFF** This hex value represents the minimum number of 4 byte locations that will be simultaneously stored in the receive RAM, relative to the beginning of the frame being input, before fabric-receive-ready signal (frrdy) may be asserted. Note that frrdy will be latent a certain amount of time due to fabric transmit clock to system transmit clock time domain crossing, and conditional on fracpt assertion. When set to maximum value, frrdy may be asserted only after the completion of the input frame. The value of this register must be greater than 18d when hstdrplt64 is asserted. The register length is shown for a receive RAM with 12 address bits (16K Bytes). **[15:0] (R/W) cfgxoffrtx: Default 0xFFFF** This hex value represents the number of pause quanta (64 bit times) after an XOFF pause frame has been acknowledged until the MAC-FIFO will reassert transmit-ControlFrame-request(tcrq) if the MAC-FIFO receive storage level has remained higher than the low watermark.
0x050	MAC-FIFO Configuration Register 2 **[31:29] Reserved** **[28:16] (R/W) cfghwm[12:0]: Default 0x1FFF** This hex value represents the maximum number of 4 byte words that will be simultaneously stored in the receive RAM before tpcf (transmit packet control frame) and psval (pause value) will facilitate an XOFF pause control frame. The register length is shown for a receive RAM with 12 address bits (16K Bytes). The register length will vary with the configured receive RAM size . **[15:13] Reserved** **[12:0] (R/W) cfglwm[12:0]: Default 0x1FFF** This hex value represents the minimum number of 4 byte words that will be simultaneously stored in the receive RAM before tpcf and psval will facilitate an XON(transmit ON) pause control frame in response to a previously transmitted XOFF pause control frame. The register length is shown for a receive RAM with 12 address bits (16K Bytes). The register length will vary with the configured receive RAM size .
0x054	MAC-FIFO Configuration Register 3 **[31:28] Reserved** **[27:16] (R/W) cfghwmft[11:0]: Default 0xFFF** This hex value represents the maximum number of 4 byte locations that will be simultaneously stored in the transmit RAM before fthwm will be asserted. Note that fthwm has two ftclk clock periods of latency before assertion or negation. This should be considered when calculating any headroom required for maximum size packets. The register length is shown for a transmit RAM with 11 address bits (8K Bytes). The register length will vary with the configured transmit RAM size. **[15:12] Reserved**

<table>
<tr><th>Address[9:0]</th><th>Function</th></tr>
<tr><td></td><td>[11:0] (R/W) cfgftth[11:0]: Default 0xFFF

This hex value represents the minimum number of 4 byte locations that will be simultaneously stored in the transmit RAM, relative to the beginning of the frame being input, before tpsf (transmit packet start of frame) will be asserted. Note that tpsf will be latent a certain amount of time due to fabric transmit clock to system transmit clock time domain crossing. When set to maximum value, tpsf will be asserted only after the completion of the input frame. The register length is shown for a transmit RAM with 11 address bits (8K Bytes). The register length will vary with the configured transmit RAM size.</td></tr>
<tr><td>0x058</td><td>MAC-FIFO Configuration Register 4

[31:18] Reserved

[17:0] (R/W) hstfltrfrm[17:0]: Default 0x0

These configuration bits are used to signal the drop frame conditions internal to the MAC-FIFO.The setting of this bits along with respective don’t care values in the hstfltrfrmdc configuration registers, create to drop the received packet by the System. For example, if it is desired to drop a frame that contains a FCS Error, bit 4 would be set.
<table>
<tr><th>Bits</th><th>Description</th></tr>
<tr><td>17</td><td>Unicast frame detected but did not match configured station address</td></tr>
<tr><td>16</td><td>Receive Frame Truncated.</td></tr>
<tr><td>15</td><td>Receive Long Event.</td></tr>
<tr><td>14</td><td>Receive VLAN Tag Detected: Frame’s length/type field contained 0x8100 which is the VLAN Protocol Identifier.</td></tr>
<tr><td>13</td><td>Receive Unsupported Op-code: Current Frame was recognized as a Control frame by the PEMCS, but it contains an unknown Op-code.</td></tr>
<tr><td>12</td><td>Receive PAUSE Control Frame: Current frame was recognized as a Control frame containing a valid PAUSE Frame Op-code and a valid address.</td></tr>
<tr><td>11</td><td>Receive Control Frame: Current Frame was recognized as a Control frame for having a valid Type-Length designation.</td></tr>
<tr><td>10</td><td>Receive Dribble Nibble: Indicates that after the end of the packet an additional 1 to 7 bits were received. A single nibble, called the dribble nibble, is formed but not sent to the system (10/100 Mbps only).</td></tr>
<tr><td>9</td><td>Receive Broadcast: Packet’s destination address contained the broadcast address.</td></tr>
<tr><td>8</td><td>Receive Multicast: Packet’s destination address contained a multicast address.</td></tr>
<tr><td>7</td><td>Receive OK: Frame contained a valid CRC and did not have a code error.</td></tr>
<tr><td>6</td><td>Receive Length Out of Range: Indicates that frame’s length was larger than 1,518 bytes but smaller than the host’s maximum frame length value (type field)</td></tr>
<tr><td>5</td><td>Receive Length Check Error: Indicates that frame length field value in the packet does not match the actual data byte length and is not a type field.</td></tr>
<tr><td>4</td><td>Receive CRC Error: Packet’s CRC did not match the internally generated CRC.</td></tr>
<tr><td>3</td><td>Receive Code Error: One or more nibbles were signalled as errors during the reception of the packet.</td></tr>
<tr><td>2</td><td>Receive False Carrier: Indicates that at some time since the last receive statistics vector, a false carrier was detected, noted, and reported with this the next receive statistics. The false carrier is not associated with this packet. False carrier is an activity on the receive channel that does not result in a packet receive attempt being made. Defined to be RXER = 1, RXDV = 0, RXD[3:0] = 0xE (RXD[7:0] = 0x0E)</td></tr>
<tr><td>1</td><td>Receive RXDV Event: Indicates that the last receive event seen was not long enough to be a valid packet.</td></tr>
<tr><td>0</td><td>Receive Previous Packet Dropped as IFG is small</td></tr>
</table></td></tr>
</table>

Address[9:0]	Function
0x05C	MAC-FIFO Configuration Register 5 **[31:23] Reserved** **[22] (R/W) cfghdplx: Default 0x0** Assertion of this bit configures the MAC-FIFO to enable half-duplex backpressure as a flow control mechanism. Deassertion of this bit configures the MAC-FIFO to enable pause frames as a flow control mechanism **[21] (RO) srfull: Default 0x0** Assertion of this read-only bit indicates that the maximum capacity of the receive FIFO storage has been met or exceeded.. **[20] (R/W) hstsrfullclr: Default 0x0** This bit should be written asserted when it is desired to clear the srfull indicator bit. After hstfullclr assertion, srfull should be read until it becomes unasserted. Hstfullclr should then be written unasserted for the indicator to become operational again. **[19] (R/W) cfgbytmode: Default 0x1** This bit should be asserted when data is transferred at the tpd and rpd bus at a rate of one byte per qualified clock. This bit should be negated when data is transferred at the tpd and rpd bus at a rate of one nibble per qualified clock. This bit should therefore be asserted when the MAC is configured for GMII mode. **[18] (R/W) hstdrplt64: Default 0x0** Setting this bit will cause the frame to be dropped if a receive frame is less than 64 bytes in length. **[17:0] (R/W) hstfltrfrmdc[17:0]: Default 0x3FFFF** The hstfltrfrmdc[17:0] configuration bits indicate which Receive Statistics Vectors are don't cares for MAC-FIFO frame drop circuitry. Setting of an hstfltrfrmdc bit, will indicate a don't care for that **hstfltrfrm** bits. Clearing the bit will look for a matching level on the corresponding hstfltrfrm bit. If a match is made then the frame is dropped.