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.

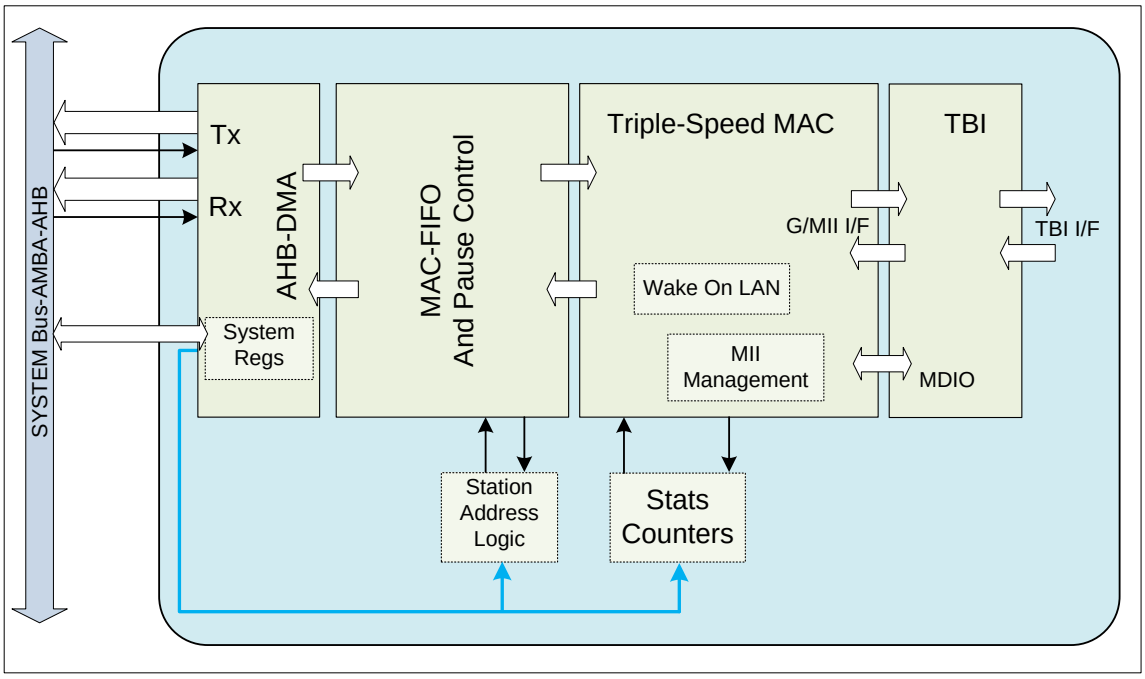


Figure 1 CoreTSE_AHB Block Diagram

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.

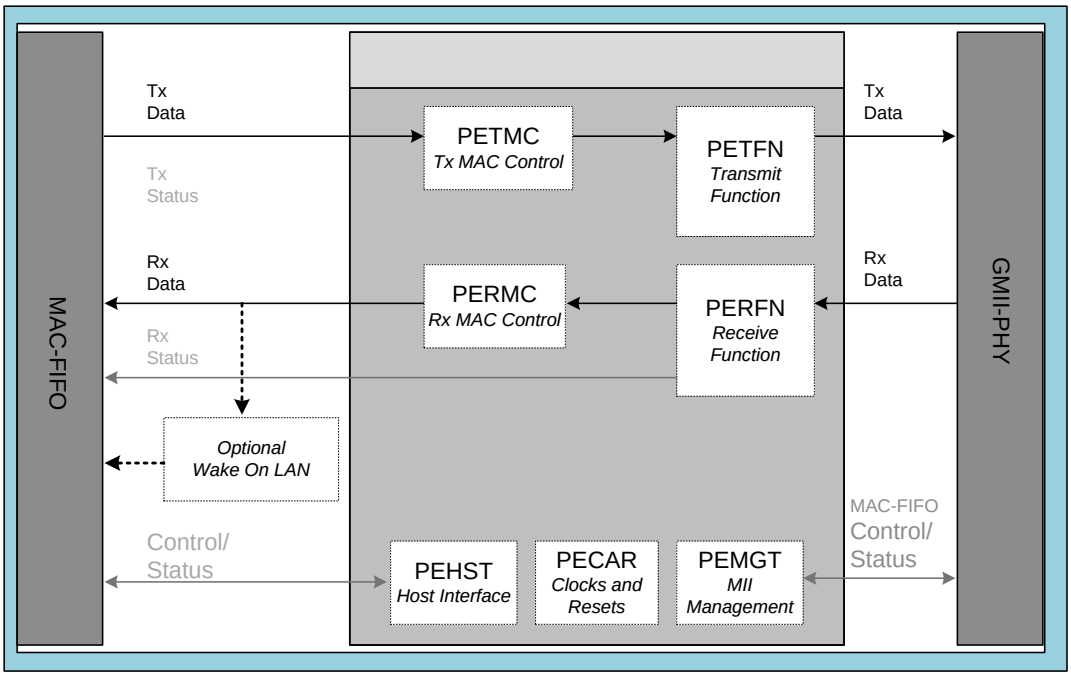


Figure 2 : Triple Speed MAC Functional Block Diagram

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

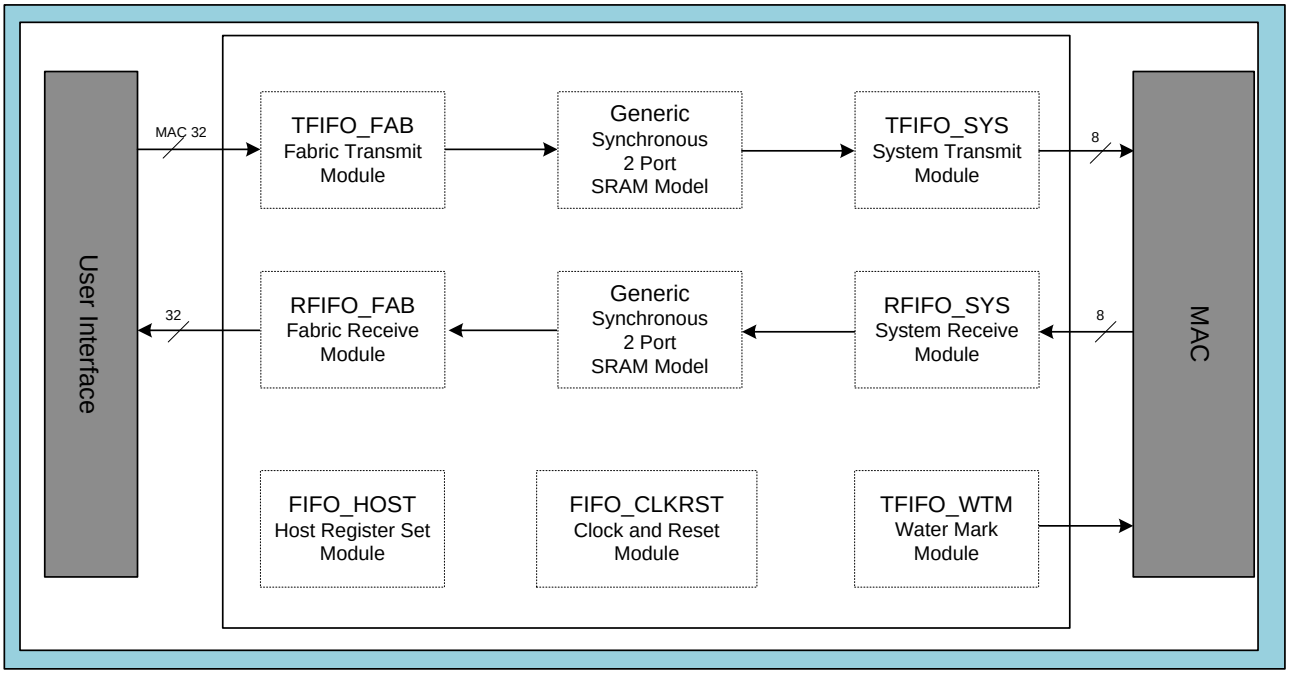


Figure 3 : MAC-FIFO Functional Block Diagram

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.

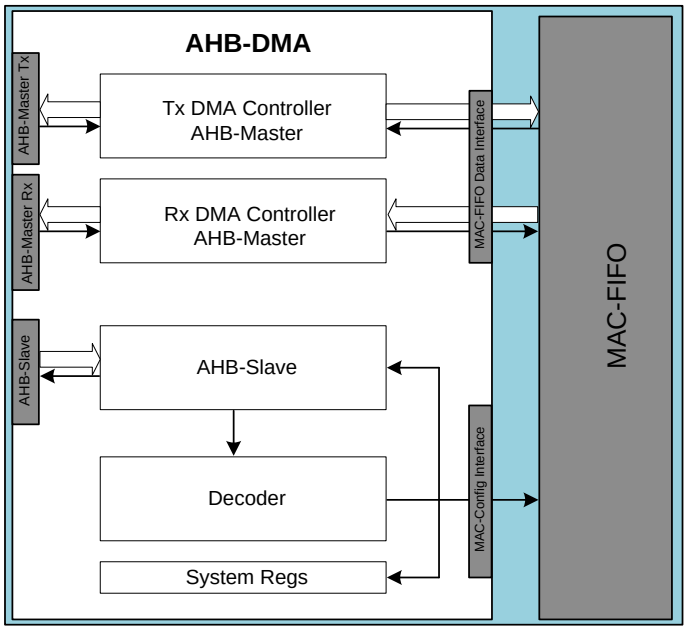


Figure 4 : AHB-DMA Functional Block Diagram

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].

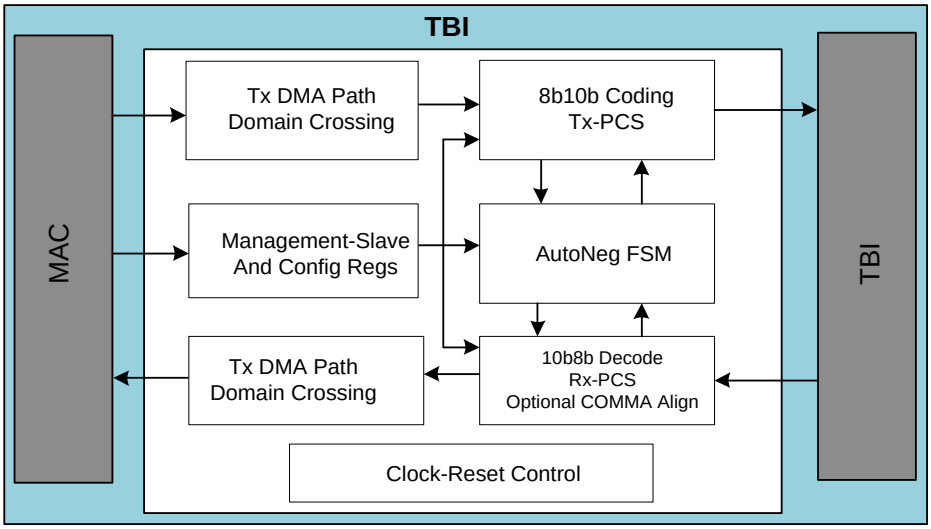


Figure 5 : TBI Functional Block Diagram

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.

Address[9:0]	Function
0x060	MAC-FIFO FIFO RAM Access* Register 0 **[31] (R/W) hsttramwreq: Default 0x0** Host transmit RAM write request. Requests the handshake of hsttramwdat and hsttramwadx values to the transmit FIFO RAM. Should only be asserted while hsttramwack is negated and while the transmit data path is disabled from receiving data and is in a steady state. It should only be negated after receiving an asserted hsttramwack **[30] (RO) hsttramwack: Default 0x0** Host transmit RAM write acknowledge. Signifies the acceptance of hsttramwdat and hsttramwadx values to the transmit FIFO RAM or TFIFO_FAB module. Will only be asserted or negated following assertion or negation of hsttramwreq.This is a read only bit. Writes specifically to this bit will have no effect. **[29:24] Reserved** **[23:16] (R/W) hsttramwdat[39:32]: Default 0x00** Host transmit RAM write data. This is the upper byte of transmit FIFO RAM data that will be written at the address of hsttramwadx[10:0] if hsttramwadx[12] is negated and hsttramwreq is asserted. This part of the transmit FIFO RAM contains control information for the frame as follows: hsttramwdat[39] = ftcfrm hsttramwdat[38:37] = ftpppadmode[1:0] hsttramwdat[36] = ftppen hsttramwdat[35] = ftppgenfcs hsttramwdat[34] = fteof hsttramwdat[33:32] = ftdatnvld[1:0] **[15:13] Reserved** **[12:0] (R/W) hsttramwadx[12:0]: Default 0x00** Host transmit RAM write address. This field has different functionality based on the value of hsttramwadx[12] and whether it is being written to or read from. When read from, hsttramwadx[11:0] field contains the actual write pointer value of the TFIFO_FAB module. When written to the hsttramwadx register will be loaded. If hsttramwadx[12] is low, hsttramwadx[10:0] will be the transmit RAM address which hsttramwdat is written to. If hsttramwadx[12] is high, hsttramwadx[11:0] contains the pointer value that will written to TFIFO_FAB module. The register definition for hsttramwadx[10:0] is for a transmit RAM with 11 address bits (8K Bytes). The register length will vary with the addressable RAM size used. The definition of hsttramwadx[12:11] will shift to remain left-justified in the register.
0x064	MAC-FIFO FIFO RAM Access* Register 1 **[31:0] (R/W) hsttramwdat[31:0]: Default 0x00** Host transmit RAM write data. This is the lower 4 bytes of transmit FIFO RAM data that will be written at the address of hsttramwadx[10:0] if hsttramwadx[12] is negated and hsttramwreq is asserted.
0x068	MAC-FIFO FIFO RAM Access* Register 2 **[31] (R/W) hsttramrreq: Default 0x0** Host transmit RAM read request. Requests the handshake of hsttramradx values to the transmit FIFO RAM and hsttramrdat from the transmit FIFO RAM. Should only be asserted while hsttramrack is negated and while the transmit data path is disabled from receiving data and is in a steady state. It should only be negated after receiving an asserted hsttramrack. **[30] (RO) hsttramrack: Default 0x0** Host transmit RAM read acknowledge. Signifies the acceptance of hsttramradx values to the transmit FIFO RAM and reception of hsttramrdat from the transmit FIFO RAM location addressed. Will only be asserted or negated following assertion or negation of hsttramrreq. This is a read only bit. Writes specifically to this bit will have no effect.

Address[9:0]	Function
	[29:24] Reserved **[23:16] (RO) hsttramrdat[39:32]: Default 0x0** Host transmit RAM read data. This is the upper byte of transmit FIFO RAM data that was read at the address of hsttramwadx[10:0] if hsttramwadx[12] is negated and hsttramwreq is asserted. This part of the transmit FIFO RAM contains control information for the frame as follows: hsttramrdat[39] = ftcfrm hsttramrdat[38:37] = ftpppadmode[1:0] hsttramrdat[36] = ftppen hsttramrdat[35] = ftppgenfcs hsttramrdat[34] = fteof hsttramrdat[33:32] = ftdatnvld[1:0] This is a read only field. Writes specifically to this field will have no effect. **[15:13] Reserved** **[12:0] (R/W) hsttramradx[12:0]: Default 0x00'** Host transmit RAM read address. If hsttramradx[12] is written low, hsttramradx[10:0] is the transmit FIFO RAM address which hsttramrdat is read from. If hsttramradx[12] is written high, hsttramradx[11:0] contains the pointer value read from TFIFO_SYS module. The register definition for hsttramradx[10:0] is for a transmit RAM with 11 address bits (8K Bytes). The register length will vary with the addressable RAM size used. The definition of hsttramradx[12:11] will shift to remain left-justified in the register.
0x06C	MAC-FIFO FIFO RAM Access* Register 3 **[31:0] (RO) hsttramrdat[31:0]: Default 0x00** Host transmit RAM read data. This is the lower 4 bytes of transmit FIFO RAM data that is read at the address of hsttramradx[10:0] if hsttramradx[12] is negated and hsttramrreq is asserted. This is a read only field. Writes specifically to this field will have no effect.
0x070	MAC-FIFO FIFO RAM Access* Register 4 **[31] (R/W) hstrramwreq: Default 0x0** Host receive RAM write request. Requests the handshake of hstrramwdat and hstrramwadx values to the receive FIFO RAM. Should only be asserted while hstrramwack is negated and while the receive data path is disabled from receiving data and is in a steady state. It should only be negated after receiving an asserted hstrramwack. **[30] (RO) hstrramwack: Default 0x0** Host receive RAM write acknowledge. Signifies the acceptance of hstrramwdat and hstrramwadx values to the receive FIFO RAM or RFIFO_SYS module. Will only be asserted or negated following assertion or negation of hstrramwreq. This is a read only bit. Writes specifically to this bit will have no effect. **[29:24] Reserved** **[23:16] (R/W) hstrramwdat[39:32]: Default 0x0** Host receive RAM write data. This is the upper byte of receive FIFO RAM data that will be written at the address of hstrramwadx[11:0] if hstrramwadx[13] is negated and hstrramwreq is asserted. This part of the receive FIFO RAM contains control information for the frame as follows: hstrramwdat[39:36] = not used hstrramwdat[35] = rpsf (receive packet start of frame) hstrramwdat[34] = rpef (receive packet end of frame) hstrramwdat[33:32] = srdatnvld[1:0]

Address[9:0]	Function
	[15:14] Reserved **[13:0] (R/W) hsttramwadx[13:0]: Default 0x0** Host receive RAM write address. This field has different functionality based on the value of hstrramwadx[13] and whether it is being written to or read from. When read from, hstrramwadx[12:0] field contains the actual write pointer value of the RFIFO_SYS module. When written to the hstrramwadx register will be loaded. If hstrramwadx[13] is low, hstrramwadx[11:0] will be the receive FIFO RAM address which hstrramwdat is written to. If hstrramwadx[13] is high, hstrramwadx[12:0] contains the pointer value that will be written to RFIFO_SYS module. The register definition for hstrramwadx[12:0] is for a receive RAM with 12 address bits (16 Kbytes). The register length varies with the addressable RAM size used. The definition of hstrramwadx[13] will shift to remain left-justified in the register.
0x074	MAC-FIFO FIFO RAM Access* Register 5 **[31:0] (R/W) hstrramwdat [31:0]: Default 0x00** Host receive RAM write data. This is the lower 4 bytes of receive FIFO RAM data that writes at the address of hstrramwadx[11:0] if hsttramwadx[13] is negated and hstrramwreq is asserted.
0x078	MAC-FIFO FIFO RAM Access* Register 6 **[31] (R/W) hstrramrreq: Default 0x0** Host receive RAM read request. Requests the handshake of hstrramradx values to the receive FIFO RAM and hstrramrdat from the receive FIFO RAM. Should only be asserted while hstrramrack is negated and while the receive data path isdisabled from receiving data and is in a steady state. It should only be negated after receiving an asserted hstrramrack. **[30] (RO) hstrramrack: Default 0x0** Host receive RAM read acknowledge. Signifies the acceptance of hstrramradx values to the receive FIFO RAM and reception of hstrramrdat from the receive FIFO RAM location addressed. Will only be asserted or negated following assertion or negation of hstrramrreq. This is a read only bit. Writes specifically to this bit will have no effect. **[29:24] Reserved** **[23:16] (RO) hstrramrdat[39:32]: Default 0x0** Host receive RAM read data. This is the upper byte of receive FIFO RAM data that was read at the address of hstrramwadx[10:0] if hstrramwadx[13] is negated and hstrramwreq is asserted. This part of the receive FIFO RAM contains control information for the frame as follows: hstrramwdat[39:36] = not used hstrramwdat[35] = rpsf hstrramwdat[34] = rpef hstrramwdat[33:32] = srdatnvld[1:0] This is a read only field. Writes specifically to this field will have no effect. **[15:14] Reserved** **[13:0] (R/W) hstrramradx[32:0]: Default 0x00** Host receive RAM read address. If hstrramradx[13] is written low, hstrramradx[11:0] is the receive FIFO RAM address which hstrramrdat is read from. If hstrramradx[13] is written high, hstrramradx[12:0] contains the pointer value read from RFIFO_FAB module. The register definition for hstrramradx[12:0] is for a receive RAM with 12 address bits (16K bytes). The register length will vary with the addressable RAM size used. The definition of hstrramradx[13] shifts to remain left-justified in the register.
0x07C	MAC-FIFO FIFO RAM Access* Register 7 **[31:0] (RO) hstrramrdat[31:0]: Default 0x00** Host receive RAM read data. This is the lower 4 bytes of receive FIFO RAM data that is read at the address of hstrramradx[11:0] if hstrramradx[13] is negated and hstrramrreq is asserted. This is a read only field. Writes specifically to this field will have no effect.

Statistics Counters Core Register

Table 12 Statistics Counters Core Register

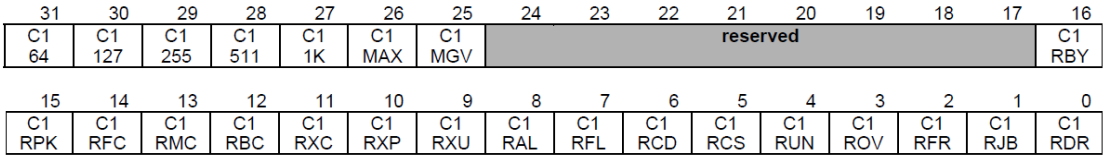
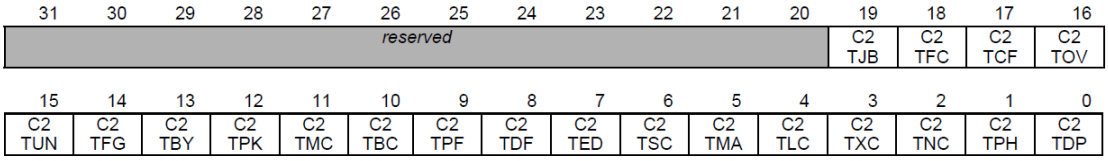
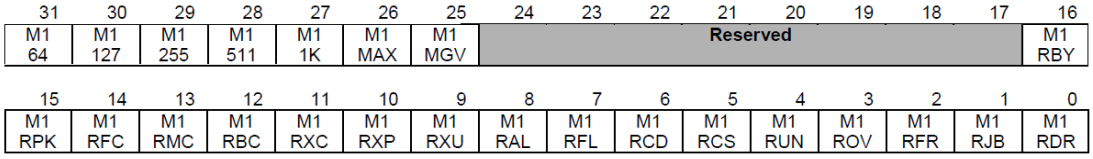
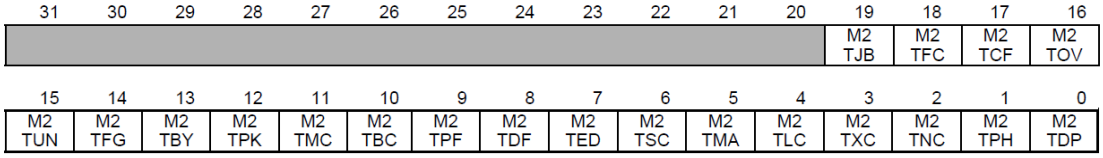
Address[9:0]	Function
0x080	TR64 - Transmit and Receive 64 Byte Frame Counter **[31:18] (R/W) Reserved** **[17:0] (R/W) Transmit and Receive 64 Byte Frame Counter** Incremented for each good or bad frame transmitted and received which is 64 bytes in length inclusive (excluding framing bits but including FCS bytes).
0x084	TR127 - Transmit and Receive 65 to 127 Byte Frame Counter **[31:18] (R/W) Reserved** **[17:0] (R/W) Transmit and Receive 65 to 127 Byte Frame Counter** Incremented for each good or bad frame transmitted and received which is 65 to 127 bytes in length inclusive (excluding framing bits but including FCS bytes).
0x088	TR255 - Transmit and Receive 128 to 255 Byte Frame Counter **[31:18] (R/W) Reserved** **[17:0] (R/W) Transmit and Receive 128 to 255 Byte Frame Counter** Incremented for each good or bad frame transmitted and received which is 128 to 255 bytes in length inclusive (excluding framing bits but including FCS bytes).
0x08C	TR511 - Transmit and Receive 256 to 511 Byte Frame Counter **[31:18] (R/W) Reserved** **[17:0] (R/W) Transmit and Receive 256 to 511 Byte Frame Counter** Incremented for each good or badframe transmitted and received which is 256 to 511 bytes in length inclusive (excluding framing bits but including FCS bytes).
0x090	TR1K - Transmit and Receive 512 to 1023 Byte Frame Counter **[31:18] (R/W) Reserved** **[17:0] (R/W) Transmit and Receive 512 to 1023 Byte Frame Counter** Incremented for each good or bad frame transmitted and received which is 512 to 1023 bytes in length inclusive (excluding framing bits but including FCS bytes).
0x094	TRMAX - Transmit and Receive 1024 to 1518 Byte Frame Counter **[31:18] (R/W) Reserved** **[17:0] (R/W) Transmit and Receive 1024 to 1518 Byte Frame Counter** Incremented for each good or bad frame transmitted and received which is 1024 to 1518 bytes in length inclusive (excluding framing bits but including FCS bytes).
0x098	TRMGV-Transmit and Receive 1519 to 1522 Byte VLAN Frame Counter **[31:18] (R/W) Reserved** **[17:0] (R/W) Transmit and Receive 1519 to 1522 Byte VLAN Frame Counter** Incremented for each good VLAN frame transmitted and received which is 1519 to 1522 bytes in length inclusive (excluding framing bits but including FCS bytes).

Address[9:0]	Function
0x09C	RBYT - Receive Byte Counter **[31:18] (R/W) Reserved** **[17:0] (R/W) Receive Byte Counter** The Statistic Counter register is incremented by the byte count of all frames received, including those in bad packets, excluding framing bits but including FCS bytes.
0x0A0	RPKT- Receive Packet Counter **[31:18] (R/W) Reserved** **[17:0] (R/W) Receive Packet Counter** Incremented for each frame received packet (including bad packets, all Unicast, Broadcast, and Multicast packets).
0x0A4	RFCS - Receive FCS Error Counter **[31:18] (R/W) Reserved** **[17:0] (R/W) Receive FCS Error Counter** Incremented for each frame received that has an integral 64 to 1518 length and contains a Frame Check Sequence error.
0x0A8	RMCA - Receive Multicast Packet Counter **[31:18] (R/W) Reserved** **[17:0] (R/W) Receive Multicast Packet Counter** Incremented for each Multicast good frame of lengths smaller than 1518 (non VLAN) or 1522 (VLAN) excluding Broadcast frames. This does not look at range/length errors.
0x0AC	RBCA - Receive Broadcast Packet Counter **[31:18] (R/W) Reserved** **[17:0] (R/W) Receive Broadcast Packet Counter** Incremented for each Broadcast good frame of lengths smaller than 1518 (non VLAN) or 1522 (VLAN) excluding Multicast frames. This does not look at range/length errors.
0x0B0	RXCF - Receive Control Frame Packet Counter **[31:18] (R/W) Reserved** **[17:0] (R/W) Receive Control Frame Packet Counter** Incremented for each MAC Control frame received (PAUSE and Unsupported).
0x0B4	RXPF - Receive PAUSE Control Frame Counter **[31:12] (R/W) Reserved** **[11:0] (R/W) Receive PAUSE Frame Packet Counter** Incremented each time a valid PAUSE MAC Control frame is received.
0x0B8	RXUO - Receive Unknown OPCode Packet Counter **[31:12] (R/W) Reserved** **[11:0] (R/W) Receive Unknown OPcode Counter** Incremented each time a MAC Control Frame is received which contains an opcode other than a PAUSE.

Address[9:0]	Function
0x0BC	RALN - Receive Alignment Error Counter **[31:12] (R/W) Reserved** **[11:0] (R/W) Receive Alignment Error Counter** Incremented for each received frame from 64 to 1518 which contains an invalid FCS and is not an integral number of bytes.
0x0C0	RFLR - Receive Frame Length Error Counter **[31:16] (R/W) Reserved** **[15:0] (R/W) Receive Frame Length Error Counter** Incremented for each frame received in which the 802.3 length field did not match the number of data bytes actually received (46 - 1500 bytes). The counter is not incremented if the length field is not a valid 802.3 length, such as an EtherType value.
0x0C4	RCDE- Receive Code Error Counter **[31:12] (R/W) Reserved** **[11:0] (R/W) Receive Code Error Counter** Incremented each time a valid carrier was present and at least one invalid data symbol was detected.
0x0C8	RCSE - Receive Carrier Sense Error Counter **[31:12] (R/W) Reserved** **[11:0] (R/W) Receive False Carrier Counter** Incremented each time a false carrier is detected during idle, as defined by a 1 on RXER and an '0xE' on RXD. The event is reported along with the statistics generated on the next received frame. Only one false carrier condition can be detected and logged between frames.
0x0CC	RUND - Receive Undersize Packet Counter **[31:12] (R/W) Reserved** **[11:0] (R/W) Receive Undersize Packet Counter** Incremented each time a frame is received which is less than 64 bytes in length and contains a valid FCS and were otherwise well formed. This does not look at Range Length errors.
0x0D0	ROVR - Receive Oversize Packet Counter **[31:12] (R/W) Reserved** **[11:0] (R/W) Receive Oversize Packet Counter** Incremented each time a frame is received which exceeded 1518 (non VLAN) or 1522 (VLAN) and contains a valid FCS and were otherwise well formed. This does not look at Range Length errors.
0x0D4	RFRG -Receive Fragments Counter **[31:12] (R/W) Reserved** **[11:0] (R/W) Receive Fragments Counter** Incremented for each frame received which is less than 64 bytes in length and contains an invalid FCS, includes integral and non-integral lengths.
0x0D8	RJBR -Receive Jabber Counter **[31:12] (R/W) Reserved** **[11:0] (R/W) Receive Jabber Counter** Incremented for frames received which exceed 1518 (non VLAN) or 1522 (VLAN) bytes and contains an invalid FCS, includes alignment errors.

Address[9:0]	Function
0x0DC	RDRP - Receive Drop **[31:12] (R/W) Reserved** **[11:0] (R/W) Receive Dropped packets Counter** Incremented for frames received which are streamed to system but are later dropped due to lack of system resources.
0x0E0	TBYT- Transmit Byte Counter **[31:24] (R/W) Reserved** **[23:0] (R/W) Transmit Byte Counter** Incremented by the number of bytes that were put on the wire including fragments of frames that were involved with collisions. This count does not include preamble/SFD or jam bytes.
0x0E4	TPKT- Transmit Packet Counter **[31:18] (R/W) Reserved** **[17:0] (R/W) Transmit Packet Counter** Incremented for each transmitted packet (including bad packets, excessive deferred packets, excessive collision packets, late collision packets, all Unicast, Broadcast, and Multicast packets).
0x0E8	TMCA- Transmit Multicast Packet Counter **[31:18] (R/W) Reserved** **[17:0] (R/W) Transmit Multicast Packet Counter** Incremented for each Multicast valid frame transmitted (excluding Broadcast frames).
0x0EC	TBCA- Transmit Broadcast Packet Counter **[31:18] (R/W) Reserved** **[17:0] (R/W) Transmit Broadcast Packet Counter** Incremented for each Broadcast frame transmitted (excluding Multicast frames).
0x0F0	TXPF- Transmit PAUSE Control Frame Counter **[31:12] (R/W) Reserved** **[11:0] (R/W) Transmit PAUSE Frame Packet Counter** Incremented each time a valid PAUSE MAC Control frame is transmitted.
0x0F4	TDFR- Transmit Deferral Packet Counter **[31:12] (R/W) Reserved** **[11:0] (R/W) Transmit Deferral Packet Counter** Incremented for each frame, which was deferred on its first transmission attempt. Does not include frames involved in collisions.
0x0F8	TEDF- Transmit Excessive Deferral Packet Counter **[31:12] (R/W) Reserved** **[11:0] (R/W) Transmit Excessive Deferral Packet Counter** Incremented for frames aborted which were deferred for an excessive period of time (3036 byte times).
0x0FC	TSCL- Transmit Single Collision Packet Counter **[31:12] (R/W) Reserved** **[11:0] (R/W) Transmit Single Collision Packet Counter** Incremented for each frame transmitted which experienced exactly one collision during transmission.

Address[9:0]	Function
0x100	TMCL- Transmit Multiple Collision Packet Counter **[31:12] (R/W) Reserved** **[11:0] (R/W) Transmit Multiple Collision Packet Counter** Incremented for each frame transmitted which experienced 2-15 collisions (including any late collisions.
0x104	TLCL- Transmit Late Collision Packet Counter **[31:12] (R/W) Reserved** **[11:0] (R/W) Transmit Late Collision Packet Counter** Incremented for each frame transmitted which experienced a late collision during a transmission attempt.
0x108	TXCL- Transmit Excessive Collision Packet Counter **[31:12] (R/W) Reserved** **[11:0] (R/W) Transmit Excessive Collision Packet Counter** Incremented for each frame that experienced 16 collisions during transmission and was aborted.
0x10C	TNCL- Transmit Total Collision Counter **[31:13] (R/W) Reserved** **[12:0] (R/W) Transmit Total Collision Counter** Incremented by the number of collisions experienced during the transmission of a frame as defined as the simultaneous presence of signals on the DO and RD circuits (i.e. transmitting and receiving at the same time).
0x110	**Not Used**
0x114	TDRP- Transmit Drop Frame Counter **[31:12] (R/W) Reserved** **[11:0] (R/W) Transmit Drop Frame Counter** Incremented each time input PFH is asserted.
0x118	TJBR- Transmit Jabber Frame Counter **[31:12] (R/W) Reserved** **[11:0] (R/W) Transmit Jabber Frame Counter** Incremented for each oversized transmitted frame with an incorrect FCS value.
0x11C	TFCS- Transmit FCS Error Counter **[31:12] (R/W) Reserved** **[11:0] (R/W) Transmit FCS Error Counter** Incremented for every valid sized packet with an incorrect FCSvalue.
0x120	TXCF- Transmit Control Frame Counter **[31:12] (R/W) Reserved** **[11:0] (R/W) Transmit Control Frame Counter** Incremented for every valid size frame with a Type Field signifying a Control frame.
0x124	TOVR- Transmit Oversize Frame Counter **[31:12] (R/W) Reserved** **[11:0] (R/W) Transmit Oversize Frame Counter** Incremented for each oversized transmitted frame with an correct FCS value.

Address[9:0]	Function
0x128	TUND- Transmit Under size Frame Counter **[31:12] (R/W) Reserved** **[11:0] (R/W) Transmit Undersize Frame Counter** Incremented for every frame less than 64 bytes, with a correct FCS value.
0x12C	TFRG- Transmit Fragments Frame Counter **[31:12] (R/W) Reserved** **[11:0] (R/W) Transmit Fragment Counter** Incremented for every frame less than 64 bytes, with an incorrect FCS value.
0x130	**CAR1 - Carry Register One [R0]** 
0x134	**CAR2 - Carry Register Two Register** 
0x138	**CAM1 - Carry Register One Mask Register** 
0x13C	**CAM2 - Carry Register Two Mask Register** 

AHB-DMA Core Registers

Table 13 AHB-DMA Core Registers

Address[9:0]	Function
0x180	DMATxCtrl - Transmit Control **[31:16] Reserved.** **[15:8] (R/W) Tx Interrupt coalescing threshold count: Default 0x1** This field is used for setting the Scatter-Gather-DMA interrupt coalescing threshold. When the Tx packet counter reaches the coalescing threshold count, interrupt bit is set in interrupt register(0x19C) **[7:1] Reserved.** **[0] (R/W) Tx Enable: Default is 0** Setting this bit enables DMA transmit packet transfers. The bit is cleared by the built-in DMA controller whenever it encounters a Tx Underrun or Bus Error state.
0x184	DMATxDescriptor - Pointer to Transmit Descriptor **[31:2] (R/W) Top 30 bits of Descriptor Address: Default '0x00'** When TxEnable is set by the host, the built-in DMA controller reads this register to discover the location in host memory of the first transmit packet descriptor. **[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
0x188	DMATxStatus - Transmit Status *Note: Flags in this register are cleared by writing a one to the bit field.* **[31:24] Reserved.** **[23:16] (R/WC) TxPktCount: Default is 0.** 8-bit transmit packet counter that is incremented whenever the built-in DMA controller successfully transfers a packet, and decremented whenever the host writes a 1 to bit 0 of this register. **[15:4] Reserved.** **[3] (R/WC) BusError: Default is 0.** When set, indicates that a host slave split, retry or error response was received by the DMA controller. **[2] Reserved.** **[1] (R/WC) TxUnderrun: Default is 0.** Set whenever the DMA controller reads a set (1) Empty Flag in the descriptor it is processing. **[0] (R/WC) TxPktSent: Default is 0.** When set, indicates that one or more packets have been successfully transferred. Writing a 1 to this bit reduces the TxPktCount value by one. The bit is cleared whenever TxPktCount is zero
0x18C	DMARxCtrl - Receive Control **[31:16] Reserved.** **[15:8] (R/W) Rx Interrupt coalescing threshold count: Default 0x1** This field is used for setting the Scatter-Gather-DMA interrupt coalescing threshold. When the Rx packet counter reaches the coalescing threshold count, interrupt bit is set in interrupt register(0x19C) **[7:1] Reserved.**

Address[9:0]	Function
	[0] (R/W) Rx Enable: Default 0 Setting this bit enables DMA receive packet transfers. When set, the built-in DMA controller will start to receive a new packet whenever the FIFO indicates that a new packet is available. The bit is cleared by the built-in DMA controller whenever it encounters an Rx Overflow or Bus Error state.
0x190	DMARxDescriptor - Pointer to Receive Descriptor **[31:2] (R/W) Top 30 bits of Descriptor Address: Default 0x00** When RxEnable is set by the host, the built-in DMA controller reads this register to discover the location in host memory of the first receive packet descriptor. **[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
0x194	DMARxStatus - Status of Rx Packet Transfers *Note: Flags in this register are cleared by writing a one to the bit field* **31:24] Reserved.** **[23:16] (R/WC) RxPktCount: Default 0** 8-bit receive packet counter that is incremented whenever the built-in DMA controller successfully transfers a packet, and is decremented whenever the host writes a 1 to bit zero of this register. **[15:4] Reserved.** **[3] (R/WC) BusError: Default 0** When set, indicates that a host slave split, retry or error response was received by the DMA controller. **[2] (R/WC) RxOverflow: Default 0** **[1] Reserved.** Set whenever the DMA controller reads a zero Empty Flag in the descriptor it is processing. **[0] (R/W) RxPktReceived: Default 0** When set, indicates that one or more packets have been successfully transferred. Writing a 1 to this bit reduces the RxPktCount value by one. The bit is cleared whenever RxPktCount is zero
0x198	DMAIntrMask Interrupt Mask *Note: Setting a bit to 1 in this register enables the corresponding status signal as an interrupt source. The DMAInterrupt register below is the AND of all DMA status bits with this mask register.* **[31:8] Reserved.** **[9] (R/W) RxPkt Interrupt coalescing Mask: Default 0** Setting this bit to '1' enables the interrupt coalescing **[8] (R/W) TxPkt Interrupt coalescing Mask: Default 0** Setting this bit to '1' enables the interrupt coalescing **[7] (R/W) Bus Error Mask: Default 0** Setting this bit to '1' enables the BusError bit in the DMARxStatus register as an interrupt source **[6] (R/W) Rx Overflow Mask: Default 0** Setting this bit to 1 enables the RxOverflow bit in the DMARxStatus register as an interrupt source **[5] Reserved.** **[4] (R/W) RxPktReceived Mask: Default 0** Setting this bit to 1 enables the RxPktReceived bit in the DMARxStatus register as an interrupt source

Address[9:0]	Function
	[3] (R/W) Bus Error Mask: Default 0 Setting this bit to 1 enables the BusError bit in the DMATxStatus register as an interrupt source **[2] Reserved.** **[1] (R/W) Tx Underrun Mask: Default 0** Setting this bit to 1 enables the TxUnderrun bit in the DMATxStatus register as an interrupt source **[0] (R/W) TxPktSent Mask: Default 0** Setting this bit to 1 enables the TxPktSent bit in the DMATxStatus register as an interrupt source
0x19C	DMAInterrupt Interrupts *Note: Read-Only register. Flags in this register are cleared when the corresponding Status bit is cleared.* **[31:8] Reserved** **[9] (RO) RxPkt Interrupt coalescing: Default 0** Set to 1 when number of RxPkt received count equals the interrupt coalescing Rx-Threshold count value. This information will result in interrupt only when the corresponding Mask bit is enabled. Once asserted, this interrupt will stay asserted till the **RxPktCount** field in RxStatus register has non-zero value. **[8] (RO) TxPkt Interrupt coalescing: Default 0** Set to 1 when number of TxPkt transmitted count equals the interrupt coalescing Tx-Threshold count value. This information will result in interrupt only when the corresponding Mask bit is enabled. Once asserted, this interrupt will stay asserted till the **TxPktCount** field in TxStatus register has non-zero value. **[7] (RO) Bus Error: Default 0** Set to 1 to record a Receive Bus Error interrupt when the BusError bit in the DMARxStatus register and bit 7 of the DMAIntrMask register are both set **[6] (RO) Rx Overflow: Default 0** Set to 1 to record an Rx Overflow interrupt when the RxOverflow bit in the DMARxStatus register and bit 6 of the DMAIntrMask register are both set **[5] Reserved.** **[4] (RO) RxPktReceived: Default 0** Set to 1 to record a RxPkt Received interrupt when the RxPktReceived bit in the DMARxStatus register and bit 4 of the DMAIntrMask register are both set **[3] (RO) Bus Error: Default 0** Set to 1 to record a Transmit Bus Error interrupt when the BusError bit in the DMATxStatus register and bit 3 of the DMAIntrMask register are both set **[2] Reserved.** **[1] (RO) Tx Underrun: Default 0** Set to 1 to record a Tx Underrun interrupt when the TxUnderrun bit in the DMATxStatus register and bit 1 of the DMAIntrMask register are both set. **[0] (RO) TxPktSent: Default 0** Set to 1 to record a Tx Pkt Sent interrupt when the TxPktSent bit in the DMATxStatus register and bit 0 of the DMAIntrMask register are both set

System Registers

Table 14 System Registers

Address[9:0]	Function
0x1C0	Frame filter controls. Default 32'h0000_003F [31:5] Reserved [5] (W/R) Pass the frame if the hash table entry matches for Multicast-DA [4] (W/R) Pass the frame if the hash table entry matches for Unicast-DA [3] (W/R) Promiscuous mode, allow all the frames to pass [2] (W/R) Pass the frame if its Unicast-DA matches the configured-DA [1] (W/R) Pass all multicast frames [0] (W/R) Pass all broadcast frames
0x1C4	(R/W) Hash Table Register0: Hash Entries-[31:0] – Default 32'h0000_0000
0x1C8	(R/W) Hash Table Register1: Hash Entries-[63:32] - Default 32'h0000_0000
0x1CC	(R/W) Hash Table Register2: Hash Entries-[95:64] - Default32'h0000_0000
0x1D0	(R/W) Hash Table Register3: Hash Entries-[127:96] - Default 32'h0000_0000 Refer the Station Address Logic for Frame Filtering section of handbook for hash table usage.
0x1D4	(R/W) miscellaneous control register - Default 32'h0000_0000

TBI/1000Base-X – Registers (Indirect Addressing through MDIO)

Configuration and status of the core is achieved by the Management Registers accessed through the serial MDIO. The TBI core is a dual mode core and to operate in TBI or 1000Base-X mode is user selectable.

The following registers are common in both modes:

- Control register at address 0x00
- Status register at address 0x01
- Extended Status register at address 0x0F
- Jitter Diagnostic register at address 0x10
- TBI Control register at address 0x11

Other registers (at address 0x04, 0x05, 0x06, 0x07, 0x08) are based on mode selected. In Table 15, these registers are described separately.

Table 15 MDIO Registers

Address	Function
0x00	**Control** **[15] (R/W, SC) PHY RESET: Default 0** Setting this bit will cause the Tx, Rx, and AutoNegX sub-modules in the TBI core to be reset. This bit is self-clearing. **[14] (R/W) LOOP BACK: Default 0** Setting this bit causes the 10-bit transmit outputs of the TBI to be connected to the receive 10-bit inputs. Clearing this bit results in normal operation. This bit does not affect the clock signals, which for loop back must be handled external to the core. **[13] Reserved. Write as 0, ignore on read.** **[12] (R/W) AUTO-NEGOTIATION ENABLE Default 0** Setting this bit enables the Auto-negotiation process. If cleared, then the values programmed determines the operating condition of the link. **[11:10] Reserved. Write as 0, ignore on read.** **[9] (R/W, SC) RESTART AUTO-NEGOTIATION: Default 0** Setting this bit causes the Auto-negotiation process to restart. This action is only available when Auto-Negotiation has been enabled. **[8:0] Reserved. Write as 0, ignore on read.**
0x01	**Status** **[15:9] Reserved. Write as 0, ignore on read.** **[8] (RO) EXTENDED STATUS: Default 1** This bit indicates that PHY status information is also contained in Register 0x0F – EXTENDED STATUS. **[7] Reserved. Write as 0, ignore on read.** **[6] (RO) MF PREMABLE SUPPRESSION ENABLE: Default 1** This bit indicates whether the PHY is capable of handling MDIO management Frames without the 32-bit preamble field. Returns 1 indicating support for suppressed preamble MDIO management Frames. **[5] (RO) AUTO-NEGOTIATION COMPLETE:** When 1, this bit indicates that the Auto-negotiation process has completed. This bit returns “0” when either the Auto-negotiation process is underway or when the Auto-negotiation function is disabled. **[4] (RO) REMOTE FAULT: Default 0** When 1, a remote fault condition has been detected between the TBI and the PHY. **[3] (RO) AUTO-NEGOTIATION ABILITY: Default 1** When 1, this bit indicates that the TBI has the ability to perform Auto-negotiation. **[2] (RO) LINK STATUS: Default 0** When 1, this bit indicates that a valid link has been established between the TBI and the PHY. When 0, no valid link has been established. **[1] Reserved. Write as 0, ignore on read.** **[0] (RO) EXTENDED CAPABILITY: Default 1** This bit indicates that the TBI contains the extended set of registers.
0x02	**Reserved**
0x03	**Reserved**

<table>
<tr><th>Address</th><th>Function</th></tr>
<tr><td>0x04</td><td>
AN Advertisement (1000BASE-T)
[15] (R/W) LINK UP:
This bit must be written 0 for TBI operation.
[14] (RO) ACK (Reserved). Ignore on read.
[13] (R/W) Reserved
This bit must be written 0 for TBI operation.
[12] (R/W) FULL-DUPLEX:
This bit must be written 0 for TBI operation.
[11:10] (R/W) LINK SPEED:
These bits must be written 00 for TBI operation.
[9:0] (R/W): These bits must always be written 0000000001 for TBI operation

AN Advertisement (1000BASE-X)
[15] (R/W) NEXT PAGE: Default 0
The local device asserts this bit to either request Next Page transmission or advertise Next Page exchange capability. This bit can thus be set when the local has no Next Pages but wishes to allow reception of Next Pages. If the local device has no Next Pages, and the Link Partner wishes to send Next Pages, the local device should send Null Message Codes and have the MESSAGE PAGE set to 0b000_0000_0001. This bit should be cleared where the local device wishes not to engage in Next Page exchange.
[14] Reserved. Write as 0, ignore on read.
[13:12] (R/W) REMOTE FAULT: Default 0x0
Encodes the local device’s remote fault condition. A fault may be indicated by setting a non-zero Remote Fault encoding and re-negotiating.
<table>
<tr><th>RF1 (4.12)</th><th>RF2 (4.13)</th><th>Description</th></tr>
<tr><td>0</td><td>0</td><td>No error, link Ok.</td></tr>
<tr><td>0</td><td>1</td><td>Offline.</td></tr>
<tr><td>1</td><td>0</td><td>Link_Failure</td></tr>
<tr><td>1</td><td>1</td><td>Auto-Negotiation_Error.</td></tr>
</table>
[11:9] Reserved
[7:8] (R/W) PAUSE: Encodes the local device’s PAUSE capability.
Pause Encoding:
<table>
<tr><th>PAUSE1 (4.7)</th><th>ASM_DIR (4.8)</th><th>Capability</th></tr>
<tr><td>0</td><td>0</td><td>No PAUSE.</td></tr>
<tr><td>0</td><td>1</td><td>Asymmetric PAUSE toward link partner.</td></tr>
<tr><td>1</td><td>0</td><td>Symmetric PAUSE.</td></tr>
<tr><td>1</td><td>1</td><td>Asymmetric PAUSE toward local device.</td></tr>
</table>
[6] (R/W) HALF-DUPLEX:
Setting this bit means local device is capable of half-duplex operation.
[5] (R/W) FULL-DUPLEX:
Setting this bit means local device is capable of full-duplex operation.
[4:0] Reserved
</td></tr>
</table>

<table>
<tr><th>Address</th><th>Function</th></tr>
<tr><td>0x5</td><td>
AN Link Partner Base Page Ability (1000BASE-T)
[15] (RO) LINK UP:
Assertion of this bit indicates that the link is up.
[14] (RO) Auto-Negotiation ACK as specified in in 802.3z
[13] Reserved
[12] (RO) FULL-DUPLEX:
Assertion of this bit indicates that the link is transferring data in Full-Duplex mode.
[11:10] (RO) LINK SPEED:
Assertion of these 2 bits indicates the speed that the link is transferring data.
2’b00: 10Mbps
2’b01: 100Mbps
2’b10: 1000Mbps
2’b11: Reserve
[9:0] (R/W): These bits must always be written 0000000001 for TBI operation.
</td></tr>
<tr><td></td><td>
AN Link Partner Base Page Ability (1000BASE-X)
[15] (RO) NEXT PAGE:
The Link Partner asserts this bit either to request Next Page transmission or to indicate the capability to receive Next Pages. When 0, the Link Partner has no subsequent Next Pages or is not capable of receiving Next Pages.
[14] (RO) ACK (Reserved): Ignore on read
[13:12] (RO) REMOTE FAULT:
<table>
<tr><th>RF1 (4.12)</th><th>RF2 (4.13)</th><th>Description</th></tr>
<tr><td>0</td><td>0</td><td>No error, link Ok.</td></tr>
<tr><td>0</td><td>1</td><td>Offline.</td></tr>
<tr><td>1</td><td>0</td><td>Link_Failure</td></tr>
<tr><td>1</td><td>1</td><td>Auto-Negotiation_Error</td></tr>
</table>
Encodes the Link Partner’s remote fault condition.
[11:9] Reserved
[8:7] (RO) PAUSE:
Encodes of the Link Partner’s PAUSE capability
Pause Encoding:
<table>
<tr><th>PAUSE (4.7)</th><th>ASM_DIR (4.8)</th><th>Capability</th></tr>
<tr><td>0</td><td>0</td><td>No PAUSE.</td></tr>
<tr><td>0</td><td>1</td><td>Asymmetric PAUSE toward link partner.</td></tr>
<tr><td>1</td><td>0</td><td>Symmetric PAUSE.</td></tr>
<tr><td>1</td><td>1</td><td>Both Symmetric PAUSE and Asymmetric PAUSE toward local device.</td></tr>
</table>
[6] (RO) HALF-DUPLEX:
When 1, Link Partner is capable of half-duplex operation.
When 0, Link Partner is incapable of half-duplex mode.
[5] (RO) FULL-DUPLEX:
When 1, Link Partner is capable of full-duplex operation. When 0, Link
Partner is incapable of full-duplex mode.
[4:0] Reserved
</td></tr>
</table>

Address	Function
0x06	**AN Expansion (1000BASE-T)** **[15:3] Reserved** **[2] (RO) NEXT PAGE ABLE: Default 1** When 1, indicates that the local device supports the Next Page function. **[1] (RO) PAGE RECEIVED:** When 1, indicates that a new page has been received and stored in the applicable AN LINK PARTNER ABILITY or AN NEXT PAGE **[0] Reserved**
	AN Expansion (1000BASE-X) **[15:3] Reserved** **[2] (RO) NEXT PAGE ABLE:** 1 indicates local device supports Next Page function. Returns 1 on read. **[1] (RO,LH) PAGE RECEIVED:** 1 indicates that a new page has been received and stored in the applicable AN LINK PARTNER ABILITY or AN NEXT PAGE register. This bit latches high in order for software to detect when polling. The bit is cleared on a read to the register. **[0] Reserved**
0x07	**AN Next Page Transmit (1000BASE-T)** Use of this register is user dependent. User can define functionality of bits of this register as per system requirement **[15:0] User defined Register**
	AN Next Page Transmit (1000BASE-X) **[15] (R/W) NEXT PAGE:** Assert this bit to indicate additional Next Pages to follow. Bit is cleared to indicate last page. **[14] (RO) ACK (Reserved): Write 0, ignore on read.** **[13] (R/W) MESSAGE PAGE:** Assert bit to indicate Message Page. Clear bit to indicate Unformatted Page. **[12] (R/W) ACKNOWLEDGE 2:** Used by Next Page function to indicate device has ability to comply with the message. Assert bit if local device will comply with message. Clear bit if local device cannot comply with message. **[11] (RO) TOGGLE:** Used to ensure synchronization with the Link Partner during Next Page exchange. This bit always takes opposite value of the Toggle bit of the previously exchanged Link Code Word. The initial value in the first Next Page transmitted is the inverse of bit 11 in the base Link Code Word. **[10:0] (R/W) MESSAGE / UNFORMATTED CODE FIELD:** Message pages are formatted pages that carry a predefined Message Code, which is enumerated in IEEE 802.3u/Annex 28C. Unformatted Code Fields take on an arbitrary value.

Address	Function
0x08	**AN Link Partner Ability Next Page (1000BASE-T)** Use of this register is user dependent. User can define functionality of bits of this register as per system requirement. **[15:0] User defined Register**
	AN Link Partner Ability Next Page (1000BASE-X) **[15] (RO) NEXT PAGE:** The Link Partner asserts this bit to indicate additional Next Pages to follow. When 0, indicates last Next Page from link partner. **[14] (RO) ACK (Reserved): Ignore on read.** **[13] (RO) MESSAGE PAGE:** When 1, indicates Message Page. When 0, indicates Unformatted Page. **[12] (RO) ACKNOWLEDGE 2:** Indicates Link Partner's ability to comply with the message. When 1, Link Partner will comply with message. When 0, Link Partner cannot comply with message. **[11] (RO) TOGGLE:** Used to ensure synchronization with the Link Partner during Next Page exchange. This bit always takes opposite value of the Toggle bit of the previously exchanged Link Code Word. The initial value in the first Next Page transmitted is the inverse of bit 11 in the base Link Code Word. **[10:0] (RO) MESSAGE / UNFORMATTED CODE FIELD:** Message pages are formatted pages that carry a predefined Message Code, which is enumerated in IEEE 802.3u/Annex 28C. Unformatted Code Fields take on an arbitrary value.
0x0F	**Extended Status** **[15] (RO) 1000BASE-X FULL-DUPLEX: Default 1** When 1, indicates PHY can operate in 1000BASE-X Full-Duplex mode. When 0, indicates PHY cannot operate in this mode. **[14] (RO) 1000BASE-X HALF-DUPLEX: Default 0** When 1, indicates PHY can operate in 1000BASE-X Half-Duplex mode. When 0, indicates PHY cannot operate in this mode. **[13] (RO) 1000BASE-T FULL-DUPLEX: Default 1** When 1, indicates PHY can operate in 1000BASE-T Full-Duplex mode. When 0, indicates PHY cannot operate in this mode. **[12] (RO) 1000BASE-T HALF-DUPLEX: Default 0** When 1, indicates PHY can operate in 1000BASE-T Half-Duplex mode. When 0, indicates PHY cannot operate in this mode. **[11:0] Reserved**

<table>
<tr><th>Address</th><th>Function</th></tr>
<tr><td>0x10</td><td>
Jitter Diagnostics
[15] (R/W) JITTER DIAGNOSTIC ENABLE: Default 0
Set this bit to enable the TBI to transmit the jitter test patterns defined in IEEE 802.3z 36A. Clear this bit to enable normal transmit-operation.
[14:12] (R/W) JITTER PATTERN SELECT: Default 0x0
<table>
<tr><th>Jitter Pattern Select</th><th>Bit 14</th><th>Bit 13</th><th>Bit 12</th></tr>
<tr><td>User Defined Custom Pattern</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>Annex 36A Defined High Frequency 10101010101010101010…</td><td>0</td><td>0</td><td>1</td></tr>
<tr><td>Annex 36A Defined Mixed Frequency 11111010110000010100…</td><td>0</td><td>1</td><td>0</td></tr>
<tr><td>Custom Defined Low Frequency 11111000001111100000…</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td>Random Jitter Pattern</td><td>1</td><td>0</td><td>0</td></tr>
<tr><td>Annex 36A Defined Low Frequency 11111000001111100000…</td><td>1</td><td>0</td><td>1</td></tr>
<tr><td>Reserved</td><td>1</td><td>1</td><td>0</td></tr>
<tr><td>Reserved</td><td>1</td><td>1</td><td>1</td></tr>
</table>
Selects the jitter pattern to be transmitted in diagnostics mode.
[11:10] Reserved
[9:0] (R/W) CUSTOM JITTER PATTERN: Default 0x0
Used in conjunction with JITTER PATTERN SELECT and JITTER DIAGNOSTIC ENABLE. Set this field to the desired custom pattern, which will be transmitted continuously.
</td></tr>
<tr><td>0x11</td><td>
Ten Bit Interface Control
[15] (R/W) SOFT RESET: Default 0
This bit resets the functional modules in the TBI. Clear it for normal operation.
[14] (R/W) SHORTCUT LINK TIMER: Default 0
Set this bit 1 to reduce the value of Go Link Timer and Sync. Status Fail Timer to 64 clock pulse. Ultimately this reduces the amount of simulation time needed to time the 1.6ms Link Timer. Clear it for normal operation. In normal operation the value of Go Link Timer is 200000 clock pulses and the value of the Sync. Status Fail Timer is 1250000 clock pulses.
[13] (R/W) DISABLE RECEIVE RUNNING DISPARITY: Default 0
Set this bit to disable the running disparity calculation and checking in the receive direction. This bit must be 0 for TBI operation.
[12] (R/W) DISABLE TRANSMIT RUNNING DISPARITY: Default 0
Set this bit to disable the running disparity calculation and checking in the transmit direction. This bit must be 0 for TBI operation.
[11] (R/W) GO LINK TIMER VALUE CONTROL: Default 0
When 0 the Go Link Timer Value=1.6ms
When set to 1 the Go Link Timer Value=10ms
[10:9] Reserved
[8] (R/W) AUTO-NEGOTIATION SENSE: Default 0
Set this bit to allow the Auto-Negotiation function to sense either a MAC in Auto-Negotiation bypass mode or an older MAC without Auto-Negotiation capability. When sensed, Auto-Negotiation Complete will become true; however Page Received will be low, indicating no page was exchanged. Management can then act accordingly. Clear this bit when IEEE 802.3z Clause 37 behaviour is desired, which results in the link not coming up.
[7:0] Reserved
</td></tr>
</table>

Interface Description

Configuration Parameters

The register transfer level (RTL) code for CoreTSE_AHB has parameters for configuring the core. While working with the core in the SmartDesign tool, a configuration GUI is used to set the values of these parameters. CoreTSE_AHB parameters are described in Table 16.

Table 16 CoreTSE_AHB Configuration Parameters

Name	Valid Range	Default	Description
FAMILY	19,24	19	Must be set to the required FPGA family: 19: SmartFusion2 24: IGLOO2
GMII_TBI	0 or 1	0	0: G/MII is active 1: TBI is active
PACKET_SIZE	256 Bytes to 32K Bytes	8K Bytes	PACKET_SIZE parameter in the design is transmit FIFO address width and supported PACKET_SIZE choices are: 256 Bytes 512 Bytes 1 K Bytes 2 K Bytes 4 K Bytes 8 K Bytes 16 K Bytes 32 K Bytes
SAL	0 or 1	1	Station Address Logic (SAL) feature 0: Disable 1: Enable
WoL	0 or 1	1	Wake on LAN (WoL) detect feature 0: Disable 1: Enable *Note: Supports Wake on LAN using AMD's Magic Packet™ Detection technology*
STATS	0 or 1	1	Statistics counters 0: Disable 1: Enable
MDIO_PHYID	0 to 31	18	MDIO Physical Address, it is an integer value. *Note: This is valid only when the TBI mode is active.*

Ports

The ports present on CoreTSE_AHB are listed in Table 17.

Table 17 CoreTSE_AHB – Signals and Descriptions

Port Name	Width	Direction	Description
Clock and Reset			
STBP	1	IN	Set Reset Bypass, used only in test-mode, where all the internal sync-resets are bypassed prior to SCAN testing. For the CoreTSE_AHB normal operation STBP must be set to '0'
TXCLK	1	IN	125/25/2.5 MHz for 1000/100/10 Mbps
RXCLK	1	IN	125/25/2.5 MHz for 1000/100/10 Mbps
GTXCLK	1	IN	125 MHz in G/MII or TBI mode
TBI PHY interface signals			
PMARX_CLK0	1	IN	62.5 MHz generated from EPCS/SERDES receive clock
PMARX_CLK1	1	IN	62.5 MHz generated from EPCS/SERDES receive clock and 180 degree out of phase with PMARX_CLK0
RCG	10	IN	Receive code group
TCG	10	OUT	Transmit code group
ANX_STATE	10	OUT	Auto negotiation status information 0th bit - DISABLE_LINK_OK state 1st bit - AN_ENABLE state 2nd bit - AN_RESTART state 3rd bit - ABILITY_DETECT state 4th bit - ACKNOWLEDGE_DETECT state 5th bit - NEXT_PAGE_WAIT state 6th bit - COMPLETE_ACKNOWLEDGE state 7th bit - IDLE_DETECT state 8th bit - LINK_OK state 9th bit - Received configuration frame data
SYNC	1	OUT	Receive link sync status
SIGNAL_DETECT	1	IN	The SIGNAL_DETECT is typically provided from the optical module to indicate when an optical signal is valid otherwise that should be driven HIGH .
TBI_READY	1	IN	External PCS Ready
TBI_TXVAL	1	OUT	TBI transmit valid
G/MII PHY interface signals			
TXD	8	OUT	Transmit Data TXD[3:0] used for MII 100/10 Mbps (Nibble Mode) TXD[7:0] used for GMII 1000 Mbps (Byte Mode)
TXEN	1	OUT	Transmit Enable
TXER	1	OUT	Transmit Error
RXD	8	IN	Receive Data RXD[3:0] used for MII 100/10 Mbps (Nibble Mode) RXD[7:0] used for GMII 1000 Mbps (Byte Mode)
RXDV	1	IN	Receive Data Valid
RXER	1	IN	Receive Error
CRS	1	IN	G/MII carrier sense flag
COL	1	IN	G/MII collision detect flag

Port Name	Width	Direction	Description
Management interface MDIO signals			
MDI	1	IN	MDIO management Data Input from pad
MDC	1	OUT	MDIO management Data Clock
MDO	1	OUT	MDIO management Data Output
MDOEN	1	OUT	MDIO management Data Output Enable
AHB-System Bus interface signals			
HRESETN	1	IN	AHB system reset(active low)
HCLK	1	IN	AHB system clock, Max 100 MHz
HREADY	1	IN	AHB ready
AHB-Slave port interface signals			
HWDATA	32	IN	AHB write data
HADDR	32	IN	AHB address
HSEL	1	IN	AHB slave select
HTRANS	2	IN	AHB transaction type
HWRITE	1	IN	AHB transaction direction
HRDATA	32	OUT	AHB slave read data
HRESP	2	OUT	AHB slave response
HREADYOUT	1	OUT	AHB slave ready
HMASTLOCK	1	IN	Locked sequence of transfers
HBURST	3	OUT	AHB Burst transfers
HSIZE	3	OUT	Size of the transfer
Tx-DMA AHB-Master interface signals			
TXHGRANT	1	IN	AHB bus grant
TXHRESP	2	IN	AHB response
TXHRDATA	32	IN	AHB read data
TXHBUSREQ	1	OUT	AHB master bus request
TXHTRANS	2	OUT	AHB master transaction type
TXHADDR	32	OUT	AHB master transaction address
TXHWRITE	1	OUT	AHB master transfer direction
TXHWDATA	32	OUT	AHB master write data bus
TXHREADY	1	IN	AHB ready, for dedicated Master-slave channel
TXHBURST	3	OUT	AHB Burst transfers
TXHLOCK	1	OUT	Locked access to the bus
TXHSIZE	3	OUT	Size of the transfer
Tx-DMA AHB-Master interface signals			
RXHGRANT	1	IN	AHB bus grant
RXHRESP	2	IN	AHB response
RXHRDATA	32	IN	AHB read data
RXHBUSREQ	1	OUT	AHB master bus request
RXHTRANS	2	OUT	AHB master transaction type

Port Name	Width	Direction	Description
RXHADDR	32	OUT	AHB master transaction address
RXHWRITE	1	OUT	AHB master transfer direction
RXHWDATA	32	OUT	AHB master write data bus
RXHREADY	1	IN	AHB ready, for dedicated Master-slave channel
RXHBURST	3	OUT	AHB Burst transfers
RXHLOCK	1	OUT	Locked access to the bus
RXHSIZE	3	OUT	Size of the transfer
Miscellaneous Signals			
TSM_INTR	3	Output	Interrupt signals. Providing these individual interrupt at top allows user to connect required interrupts to host-processor based on application requirement. [2] Wake On LAN detected interrupt [1] Statistics counter carry interrupt [0] DMA interrupt
TSM_CONTROL	32	Output	32bit GPIO output signals mapped to system miscellaneous control register (0x1D4)

Licensing

CoreTSE_AHB is licensed as obfuscated register transfer level (RTL).

RTL

Complete obfuscated RTL source code is provided for the core.

SmartDesign

CoreTSE_AHB is available through the Libero SoC IP Catalog. Download it from a remote web-based repository and install into your local vault to make it ready to use. Once installed in the Libero software, the core can be instantiated, configured, connected, and generated using the SmartDesign tool.

An example instantiated view is shown in Figure 6.

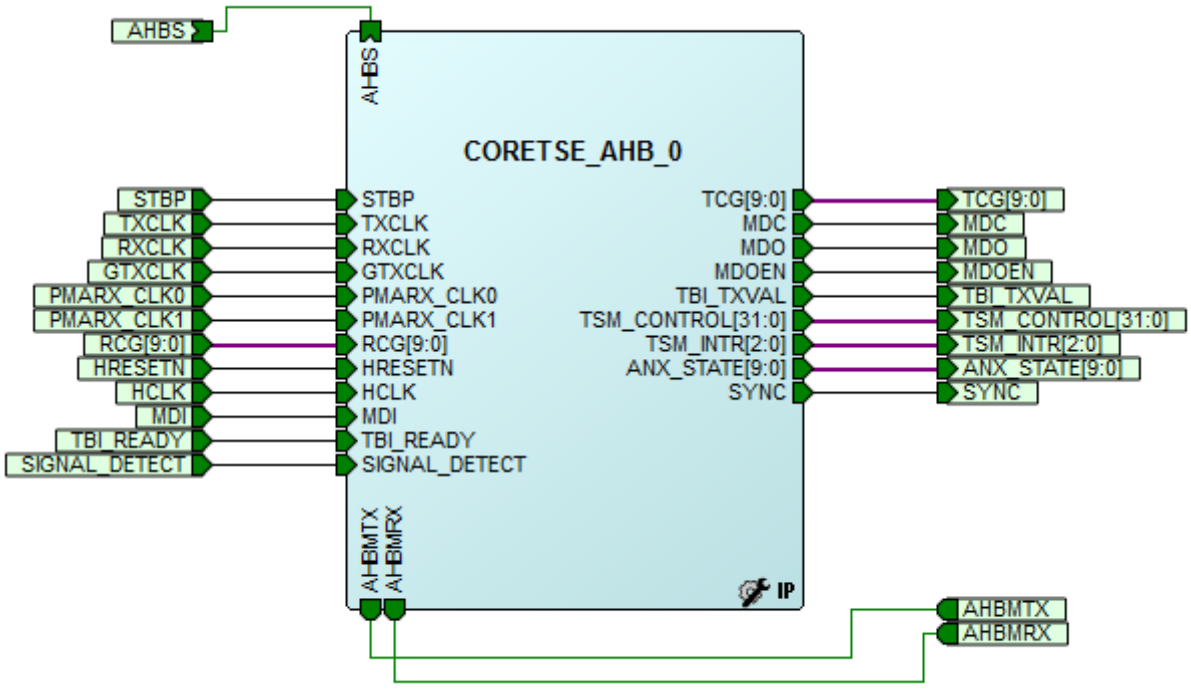


Figure 6 SmartDesign CoreTSE_AHB instance view

For more information on using SmartDesign to instantiate and generate cores, refer to the *Using DirectCore in Libero® System-on-Chip (SoC) User Guide* or consult the Libero SoC online help.

Configuring CoreTSE_AHB in SmartDesign

The CoreTSE_AHB configuration GUI takes up a large amount of screen area when it is sized to show all configuration options. Figure 7 shows the top portion of the configuration GUI.

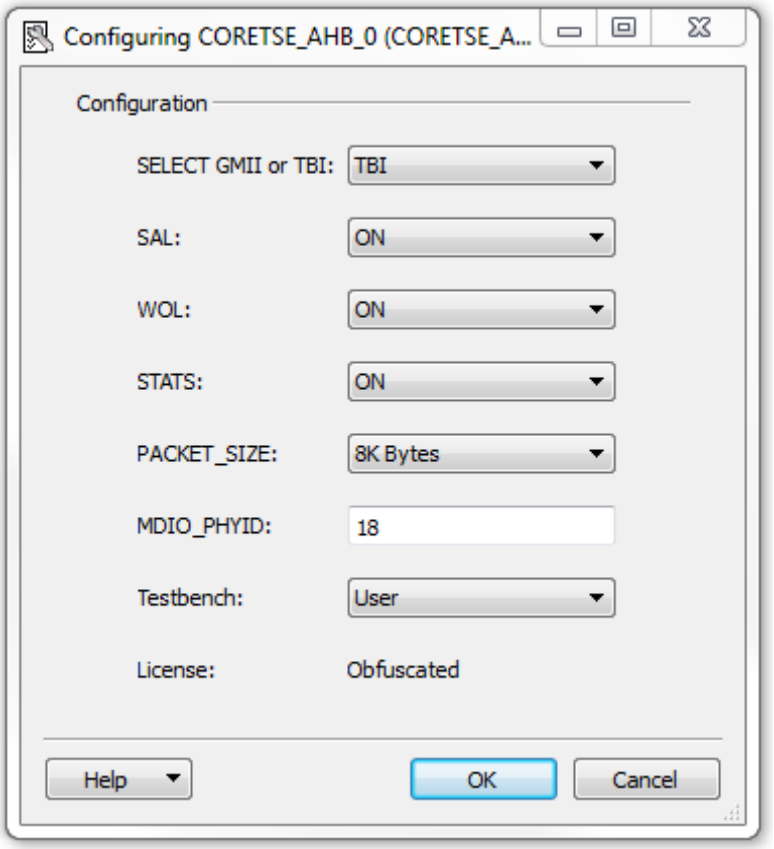


Figure 7 CoreTSE_AHB Configuration GUI (Top Portion)

Testbench Operation and Modification

A unified testbench is used to verify and test CoreTSE_AHB. It is called a user testbench.

User Testbench

A simplified block diagram of the user testbench is shown in Figure 8. The user testbench instantiates the CoreTSE_AHB as well as behavioral, non-synthesizable models of an input test generator and provides necessary clock, reset, and other signals. The testbench compares the actual CoreTSE_AHB output interface against the predicted behavioral model data.

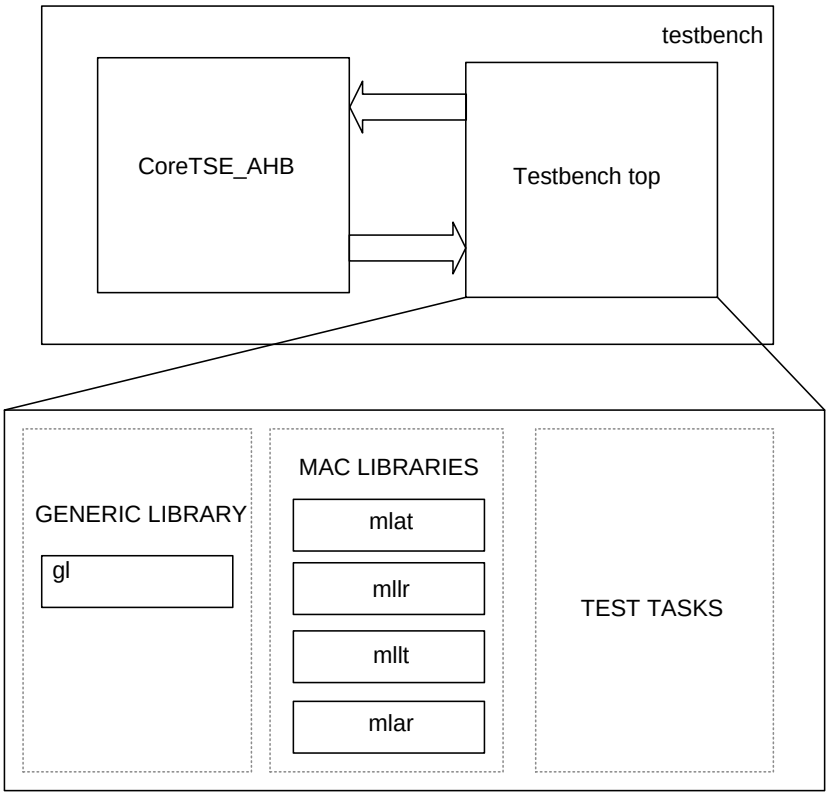


Figure 8 CoreTSE_AHB User Testbench

The same testbench can be used for pre-synthesis and post-synthesis simulation. A simulation tool displays the verification result. Test bench has task based library models for AHB transmit, AHB receive, MAC link transmit, MAC link receive, and generic test bench to check and report errors. Following are the modules instances used in testbench top:

- gl - Generic Library instantiations
- mlat - MAC library for AHB transmit
- mllr - MAC library for link receive
- mllt - MAC library for link transmit
- mlar - MAC library for AHB receive
- MAC library module has tasks related to MAC functionality
  - Tasks for generating descriptors
  - Tasks related to 8bto10b conversation and 10bto8b conversion logic
- Module gl has tasks related to generic test bench functionality
  - Tasks for simulation end, error message prints and maximum error definitions.

In TBI mode, the following test case is available:

1. Random and data, IPG, and preamble pattern test
    - This test DMA's random size and payload data frames through Transmit and then receive paths of the CoreTSE_AHB.

In G/MII mode, the following test cases are available:

1. Random Data and Timing test
    - This test DMA's random size and payload data frames through transmit and then receive paths of the CoreTSE_AHB. This is done for different speed modes.
2. Per Packet configuration test
    - This test enables Per Packet configuration of transmit path data and then verifies the operation of all bits at the link.

System Integration

This section provides hints to ease the integration of CoreTSE_AHB.

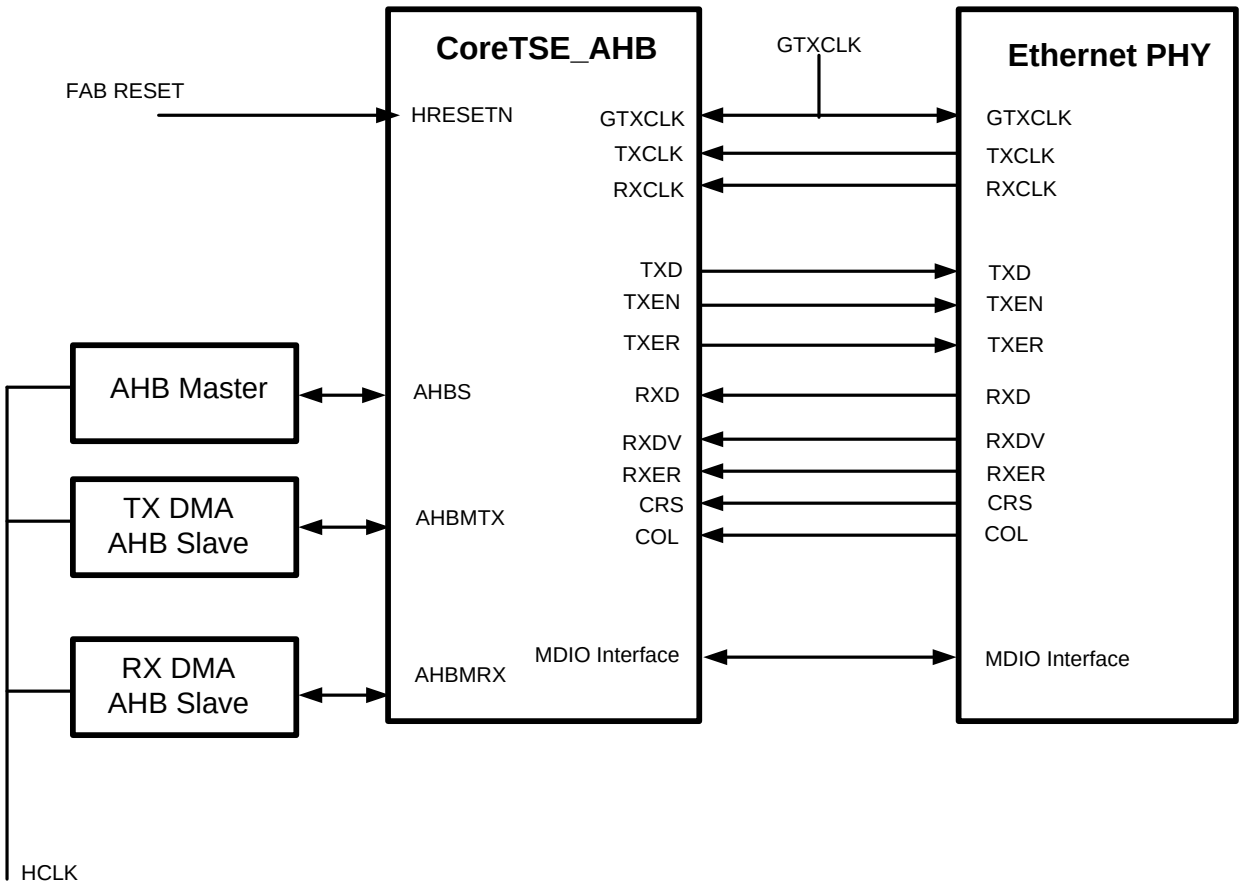


Figure 9 CoreTSE_AHB Integration in G/MII System

- Fabric reset used for CoreTSE_AHB resets.
- RXCLK is 125/25/2.5 MHz for 1000/100/10 Mbps respectively and they are driven from Ethernet PHY.
- TXCLK is 25/2.5 MHz for 100/10 Mbps respectively and they are driven from Ethernet PHY.
- GTXCLK is 125 MHz clock for 1000 Mbps.
- 50 MHz HCLK is used for the AHB subsystem in the core is driven from the application host clock.

Run the Libero flow with enabling the Timing Driven and High Effort Place and Route option Example design clock constraints are included in the core package, generated under smart design path

/component/Actel/DirectCore/CORETSE_AHB/ <Core version number>/constraints/

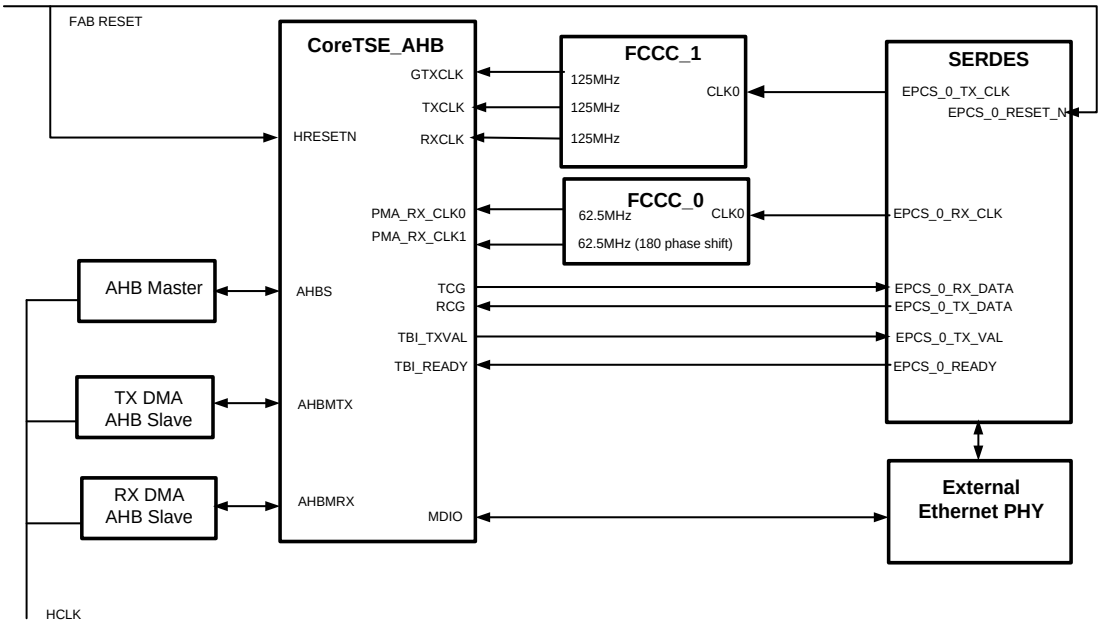


Figure 10 CoreTSE_AHB Integration in TBI System

- The design created to use the TBI interface for 1000Base-T in 1000Mbps mode only.
- Fabric reset used for CoreTSE_AHB and SERDES resets.
- PMARX_CLK0 and PMARX_CLK1 are 62.5 MHz clocks and they are 180 degree out of phase each other, these clocks are generated from FCCC_0. Input clock for the FCCC_0 is 125 MHz (EPCS_0_RX_CLK) from the SERDES.
- TXCLK, RXCLK, and GTXCLK are 125 MHz clocks generated from FCCC_1. The input for the FCCC_1 is 125 MHz from EPCS_0_TX_CLK from the SERDES.
- 50 MHz HCLK is required for the AHB subsystem in the core can be driven from the application host clock.
- Run the Libero flow with enabling the Timing Driven and High Effort Place and Route option

Example design clock constraints are included in the core package, generated under smart design path /component/Actel/DirectCore/CORETSE_AHB/ < Core version number> /constraints/

Ordering Information

Ordering Codes

CoreTSE_AHB v2.1 can be ordered through Microsemi® local Sales Representative. It should be ordered using the following number scheme: CoreTSE_AHB-XX, where XX is listed in Table 18.

Table 18 Ordering Codes

XX	Description
OM	RTL for obfuscated RTL source — multiple-use license

List of Changes

The following table shows important changes made in this document for each revision.

Revision*	Changes	Page
Revision 2 (March 2015)	CoreTSE_AHB v2.1 release.	N/A
Revision 1 (August 2014)	CoreTSE_AHB v2.0 release.	N/A

*Note: *The revision number is located in the part number after the hyphen. The part number is displayed at the bottom of the last page of the document. The digits following the slash indicate the month and year of publication.*

Product Support

Microsemi SoC Products Group backs its products with various support services, including Customer Service, Customer Technical Support Center, a website, electronic mail, and worldwide sales offices. This appendix contains information about contacting Microsemi SoC Products Group and using these support services.

Customer Service

Contact Customer Service for non-technical product support, such as product pricing, product upgrades, update information, order status, and authorization.

From North America, call **800.262.1060**
From the rest of the world, call **650.318.4460**
Fax, from anywhere in the world **650. 318.8044**

Customer Technical Support Center

Microsemi SoC Products Group staffs its Customer Technical Support Center with highly skilled engineers who can help answer your hardware, software, and design questions about Microsemi SoC Products. The Customer Technical Support Center spends a great deal of time creating application notes, answers to common design cycle questions, documentation of known issues and various FAQs. So, before you contact us, please visit our online resources. It is very likely we have already answered your questions.

Technical Support

For Microsemi SoC Products Support, visit http://www.microsemi.com/products/fpga-soc/design-support/fpga-soc-support.

Website

You can browse a variety of technical and non-technical information on the Microsemi SoC Products Group home page, at http://www.microsemi.com/soc/.

Contacting the Customer Technical Support Center

Highly skilled engineers staff the Technical Support Center. The Technical Support Center can be contacted by email or through the Microsemi SoC Products Group website.

Email

You can communicate your technical questions to our email address and receive answers back by email, fax, or phone. Also, if you have design problems, you can email your design files to receive assistance. We constantly monitor the email account throughout the day. When sending your request to us, please be sure to include your full name, company name, and your contact information for efficient processing of your request.

The technical support email address is soc_tech@microsemi.com.

My Cases

Microsemi SoC Products Group customers may submit and track technical cases online by going to My Cases.

Outside the U.S.

Customers needing assistance outside the US time zones can either contact technical support via email (soc_tech@microsemi.com) or contact a local sales office. Sales office listings can be found at www.microsemi.com/soc/company/contact/default.aspx.

ITAR Technical Support

For technical support on RH and RT FPGAs that are regulated by International Traffic in Arms Regulations (ITAR), contact us via soc_tech_itar@microsemi.com. Alternatively, within My Cases, select **Yes** in the ITAR drop-down list. For a complete list of ITAR-regulated Microsemi FPGAs, visit the ITAR web page.

Microsemi Corporate Headquarters
One Enterprise, Aliso Viejo,
CA 92656 USA

Within the USA: +1 (800) 713-4113
Outside the USA: +1 (949) 380-6100
Sales: +1 (949) 380-6136
Fax: +1 (949) 215-4996

E-mail: sales.support@microsemi.com

Microsemi Corporation (Nasdaq: MSCC) offers a comprehensive portfolio of semiconductor and system solutions for communications, defense & security, aerospace and industrial markets. Products include high-performance and radiation-hardened analog mixed-signal integrated circuits, FPGAs, SoCs and ASICs; power management products; timing and synchronization devices and precise time solutions, setting the world's standard for time; voice processing devices; RF solutions; discrete components; security technologies and scalable anti-tamper products; Power-over-Ethernet ICs and midspans; as well as custom design capabilities and services. Microsemi is headquartered in Aliso Viejo, Calif., and has approximately 3,400 employees globally. Learn more at **www.microsemi.com**.

50200571-2/03.15