PLB TEMAC (v2.00a)

DS489 June 30, 2005 **Product Specification**

Introduction

This document provides the design specification for the PLB_TEMAC (PLB interfaced Tri-mode Ethernet Media Access Controller). Tri-mode indicates that this core may transmit and receive data at three rates, 10, 100, or 1000 Megabits per second (Mb/s).

The PLB_TEMAC described in this document (when combined with the Hard TEMAC core) has been designed incorporating the applicable features described in IEEE Std. 802.3-2002. Differences between that specification and the Xilinx PLB_TEMAC implementation are highlighted and explained in Specification Exceptions.

The PLB_TEMAC is an intellectual property (IP) core designed for implementation in a Virtex-4 FX FPGA. The PLB_TEMAC provides an interface between the Processor Local Bus (PLB) and one half of the Hard Tri-mode Ethernet Media Access Controller (Hard TEMAC) core implemented in the FPGA silicon. This connection must be accomplished by building an embedded system with the Embedded Development Kit (EDK) tools.

The PLB_TEMAC v2.00a core is designed to be used with the HARD_TEMAC v1.00a core.

Two instances of the PLB_TEMAC are required if both halves of the Hard TEMAC will be used. Those Virtex-4 FX FPGAs which have two PowerPC™ processors have a second Hard TEMAC available. Each half of a Hard TEMAC to be used requires a separate instance of the PLB_TEMAC.

The PLB_TEMAC communicates to a processor via a 64-bit Processor Local Bus (PLB) using the Intellectual Property InterFace (PLB_IPIF). The PLB_IPIF interfaces with registers in the IPIF itself, intermediate blocks in the PLB_TEMAC and, registers within the Hard TEMAC.

Future releases of the PLB_TEMAC will support an option to access the registers in the Hard TEMACs through the Device Control Register bus (DCR).

LogiCORE™ Facts		
Core Specifics		
Supported Device Family	Virtex-4™-FX	
Version of Core	plb_temac	v2.00a
Resources Used		
	Min	Max
Total Core I/O	884	890
Core FPGA IOBs	0	0
LUTs	1492	4481
FFs	1145	2414
Block RAMs	4	32
Provided with Core		
Documentation	Product Specification	
Design File Formats	VHDL	
Constraints File	N/A	
Verification	N/A	
Instantiation Template	N/A	
Reference Designs	None	
Design Tool Requirements		
Xilinx Implementation Tools	7.1.03i or later	
Verification	N/A	
Simulation	ModelSim SE/EE 5.7b or later	
Synthesis	XST	
Support		
Support provided by Xilinx, Inc.		

The Xilinx PLB_TEMAC may be configured for particular systems by appropriately selecting features through parameter values. This allows a user to craft a design with the minimum necessary resources while providing the desired operational functions. Parameterizable features of the design are discussed in PLB_TEMAC Design Parameters.

Tri-mode EMAC System Overview

A Tri-mode Ethernet System includes either one or two PLB_TEMAC cores for each HARD_TEMAC core that will be used. **This release of the PLB_TEMAC does not support a two PLB_TEMAC system[1].**

HARD_TEMAC Core

The Hard TEMAC is a silicon component of each Virtex™4 FX FPGA. The HARD_TEMAC core enables the use of the Hard TEMAC silicon component in EDK embedded systems. Each Hard TEMAC consists of two independent Ethernet Medium Access Controllers (EMAC) capable of 10, 100, or 1000 Mb/s communications and complies with IEEE 802.3-2002 specifications.

These EMACs may be configured for full or half duplex operation and support several media interfaces including MII, GMII, RGMII, SGMII, and 1000Base-X. The Hard TEMAC also supports MII management of physical devices, PHY, VLAN frames[2], jumbo frames, configurable inter-frame gaps, in-band frame check sequences, FCS, for both transmit and receive, auto padding on transmit, FCS stripping on receive, flow control through Pause packets, receive address filtering, and provides raw statistics vector outputs.

This consumes no FPGA resources since the Hard TEMAC is built into the silicon of each Virtex™4 FX FPGA. Please refer to the HARD_TEMAC core specification for more details specifically about the function of the Hard TEMAC.

Figure 1 is a block diagram of the Hard TEMAC and the PowerPC processor Silicon Components.

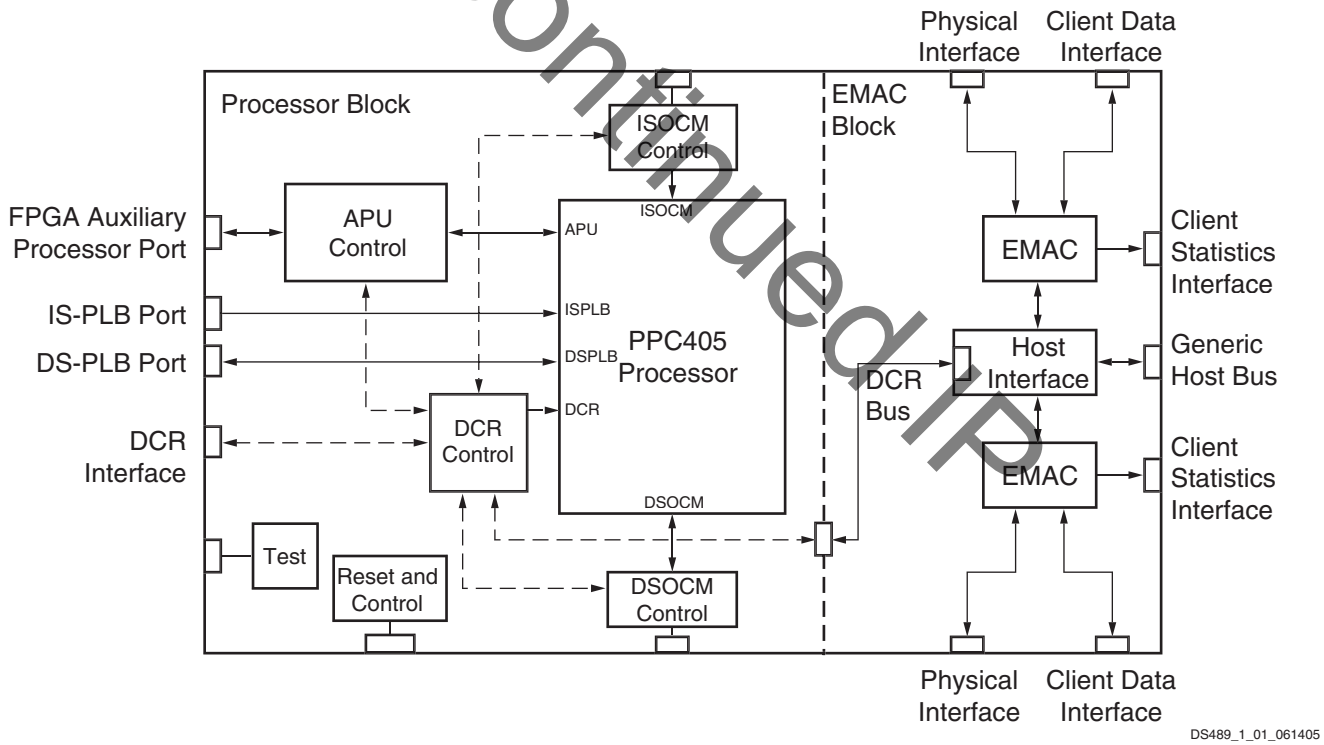

Figure 1: **Block Diagram of the Hard TEMAC and PowerPC Silicon Components**

1. Throughout this specification, features will be described as either "Not supported in this release" or "Not implemented in this release". Items designated as "Not supported" are included in the design and may have been tested with simulation but have not been tested in hardware and have not been integrated with the software drivers. These functions may be functional if used but are not supported at this time. Items that are designated "Not implemented" are not included in the design at all and will not work in this release.
2. IEEE Std. 802.3 uses the terms Frame and Packet interchangeably when referring to the Ethernet unit of transmission.

Figure 2 is a detailed block diagram of the Hard TEMAC. This shows two EMACs with a unified Host interface for access to the configuration registers of both EMACs. The Host interface can be accessed either from a generic signal interface or from a DCR connection which is part of the hard silicon design of the PowerPC and Hard TEMAC. Each EMAC has its own set of Client, PHY, PHY Management, and statistics interfaces.

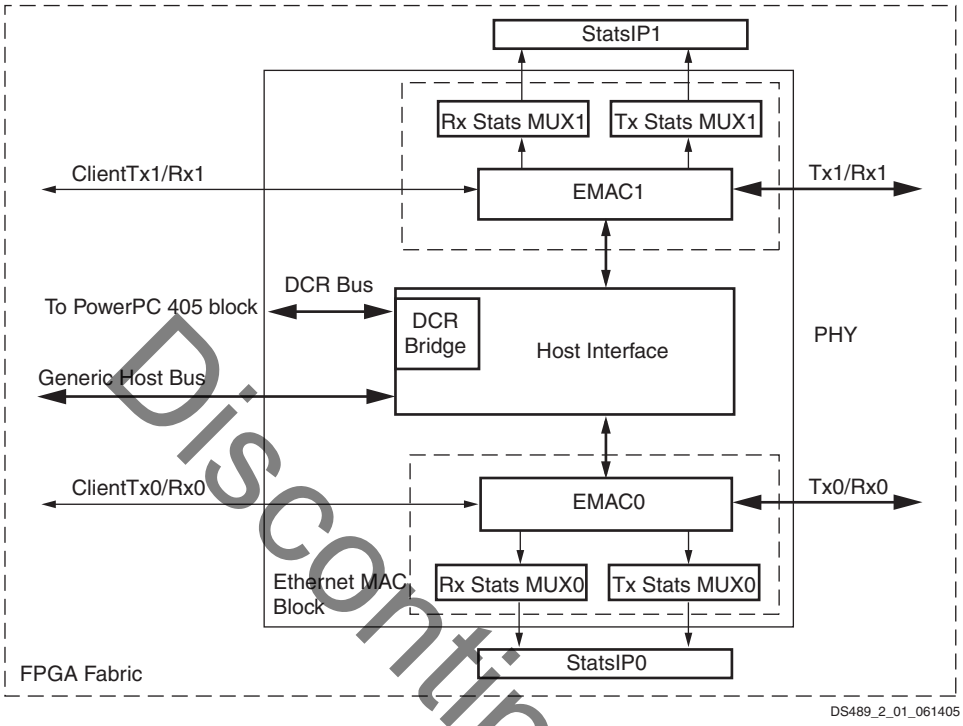

Figure 2: **Detailed Block Diagram of the Hard TEMAC Silicon Component**

PLB_TEMAC Core

The PLB_TEMAC provides access to the HARD_TEMAC host interface from the PLB. Alternately, the DCR Interface, built in the silicon, may be used for host access. This PLB_TEMAC release does not support host access through the DCR.

The PLB_TEMAC enables memory mapped access to registers and memory mapped or DMA access to packet FIFOs which in turn interface to the Client transmit and receive interfaces of the HARD_TEMAC to support transmission and reception of Ethernet frames. The PLB_TEMAC is comprised of several blocks as shown in Figure 3.

Although there are separate specifications for the PLB_IPIF and the HARD_TEMAC, this specification addresses the specific implementation required for the PLB_TEMAC use.

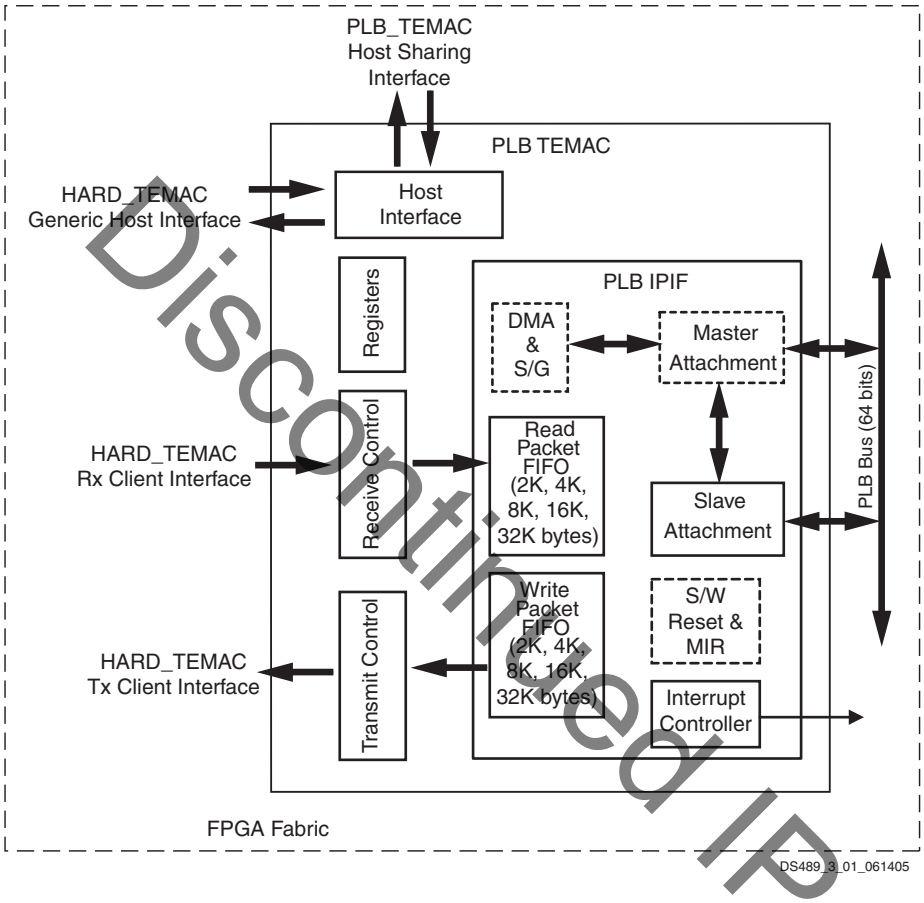

Figure 3: **PLB TEMAC Block Diagram**

PLB_TEMAC Endianess

Note that the PLB_TEMAC is designed as a big endian device (bit 0 is the most significant bit and is shown on the left of a group of bits). The Hard TEMAC is designed as a little endian device (bit 0 is the least significant bit and is shown on right of a group of bits).

The 8-bit GMII transmit and receive data interface to the external PHY is little endian (bit 7 is the most significant bit and appears on the left of the bus). The MII management interface to the PHY is serial with the most significant bit of a field being transmitted first.

Features

The PLB_TEMAC is an IP core designed for Xilinx FPGAs and contains the following features:

- 64-bit PLB master and slave interfaces.[1]
- Memory mapped direct I/O interface to registers and FIFOs as well as Simple DMA and Scatter/Gather DMA capabilities for lower processor and bus utilization.
- Independent internal 2K, 4K, 8K, 16K, or 32K byte TX and RX FIFOs for holding data for more than one frame (2K byte depth is sufficient for one normal 1518 maximum byte frame but 4K byte depth provides better throughput. 16K or 32K byte depth is required for Jumbo frames up to 9K bytes long).
- 16, 32 or 64 entry deep FIFOs for the Transmit Length, Receive Length, and Transmit Status registers to support multiple frame operation.
- Filtering of “bad” receive frames to reduce processor bus utilization.
- Hardware selectable DCR or PLB host interface to configuration registers. **(DCR not supported in this release)**.
- RGMII/GMII/MII interface to external PHY devices. (**RGMII & MII not supported in this release).**
- SGMII supported through MGT interface to external copper PHY layer. **(SGMII not supported in this release).**
- Complies with IEEE 802.3-2000 specification.
- Configurable full or half duplex operation. **(Half duplex not implemented in this release).**
- Media Independent Interface Management (MIIM) for access to PHY transceiver registers. **(MIIM not implemented in this release).**
- Optional 1000 Base -X PCS/PMA sublayer for complete on-chip 1000 Base-X implementation when used with on-chip MGTs. **(1000 Base -X not supported in this release).**
- Auto pad and Frame Check Sequence (FCS) field insertion or pass through on transmit.
- Auto pad and FCS field stripping or pass through on receive. **(Pad & FCS stripping not implemented in this release).**
- Processes transmission and reception of Pause packets for flow control.
- Supports receive and transmit of longer VLAN type frames compliant to IEEE 802.3-2000.
- Programmable interframe gap.
- Provides interrupts for many error and status conditions.
- Optional support of jumbo frames up to 9K bytes in length.

PLB_TEMAC Design Parameters

To allow the user to generate an PLB_TEMAC that is tailored for their system, certain features are parameterizable in the PLB_TEMAC design. This allows the user to have a design that only utilizes the resources required by their system and runs

1. The PLB master interface is only used if either simple or scatter gather DMA is included in the core at build time. The core always includes a PLB slave interface.

at the best possible performance. The features that are parameterizable in the Xilinx PLB_TEMAC design are shown in Table 1.

Table 1: **PLB_TEMAC Design Parameters**

Grouping / Number		Feature / Description	Parameter Name	Allowable Values	Default Value	VHDL Type
Top Level	G1	Device Block Id	C_DEV_BLK_ID	0 to 255	1	integer
	G2	BUS clock period in pS	C_PLB_CLK_PERIOD_PS	Requirements as stated in note 1	8000	integer
	G3	Device family	C_FAMILY	Virtex4	Virtex4	string
	G4	FIFO depth in bits	C_IPIF_WRFIFO_DEPTH	262144,131072, 65536, 32768, 16384	32768	integer
	G5	FIFO depth in bits	C_IPIF_RDFIFO_DEPTH	262144,131072, 65536, 32768, 16384	32768	integer
	G6	MAC length and status FIFO depth	C_MAC_FIFO_DEPTH	16, 32, 64	16	integer
	G7	Device base address	C_BASEADDR	See note 2	FFFFFFFF	std logic vector
	G8	Device maximum address	C_HIGHADDR	See note 2	0x0	std logic vector
PLB/IPIF Interface	G9	Module Identification Read	C_INCLUDE_DEV_MIR	1 = MIR reads Exists 0 = MIR reads Non-existent	1	integer
	G10	Data Re-alignment Engine	C_INCLUDE_DRE	1 = DRE Exists 0 = DRE Non-existent	1	integer
	G11	Software Reset Function	C_INCLUDE_RESET	1 = software reset Exists 0 = software reset Non-existent	1	integer
	G12	Interrupt device ID encoder	C_INCLUDE_DEV_PENCODER	1 = interrupt device ID encoder Exists 0 = interrupt device ID encoder Non-existent	1	integer
	G13	DMA Type	C_DMA_TYPE(3)	1 = no DMA function is required 2 = simple 2 channel DMA is required 3 = Scatter Gather DMA for packets is required	3	integer

Notes:

1. The PLB BUS clock frequency must be greater than or equal to 42 MHz and less than or equal to 125 MHz for 1 Gb/s Ethernet operation.
2. The default value will insure that the actual value is set, i.e., if the value is not set, a compiler error will be generated. The address range must be at least 0x4000 (for example, 0x10000000 and 0x10003FFF). C_BASEADDR must be a multiple of the range, where the range is C_HIGHADDR - C_BASEADDR +1.
3. When C_DMA_TYPE is '2' or '3' a PLB master interface is included in the core. When C_DMA_TYPE is '1', no PLB master interface is used. The PLB slave interface is always present.

Table 1: **PLB_TEMAC Design Parameters *(Continued)***

Grouping / Number		Feature / Description	Parameter Name	Allowable Values	Default Value	VHDL Type
	G14	DMA interrupt coalescing functionality	C_INCLUDE_DMA_INTR_COASLESCE	1	1	integer
	G15	PLB number of masters	C_PLB_NUM_MASTERS	The number of Master Devices connected to the PLB bus	8	integer
	G16	PLB master ID width	C_PLB_MID_WIDTH	The width of the Master ID bus. This is set to roundup(log2 (C_PLB_NUM_MASTERS))	3	integer
	G17	PLB address bus width (in bits)	C_PLB_AWIDTH	32	32	integer
	G18	PLB data bus width (in bits)	C_PLB_DWIDTH	64	64	integer
TEMAC Features	G19	EMAC hard (built in Virtex4 silicon) or soft (built in FPGA fabric)	C_TEMAC_IS_SOFT	0 = hard 1 = soft **(Only 0 is implemented in this release).**	0	integer
	G20	EMAC host interface is Memory Mapped or accessed through DCR	C_TEMAC_DCR_HOST	0 = Memory Mapped 1 =DCR **(Only 0 is supported in this release).**	0	integer
	G21	Each processor block has two EMAC instances. Which EMAC instance is used here?	C_TEMAC_INST	0 = instance 0 1 = instance 1 **(Only 0 is supported in this release).**	0	integer
	G22	How many hard EMACs from this processor block are used	C_TEMAC_BOTH_USED	0 = 1 instance is used 1 = both instances used **(Only 0 is supported in this release).**	0	integer

Notes:
1. The PLB BUS clock frequency must be greater than or equal to 42 MHz and less than or equal to 125 MHz for 1 Gb/s Ethernet operation.
2. The default value will insure that the actual value is set, i.e., if the value is not set, a compiler error will be generated. The address range must be at least 0x4000 (for example, 0x10000000 and 0x10003FFF). C_BASEADDR must be a multiple of the range, where the range is C_HIGHADDR - C_BASEADDR +1.
3. When C_DMA_TYPE is '2' or '3' a PLB master interface is included in the core. When C_DMA_TYPE is '1', no PLB master interface is used. The PLB slave interface is always present.

Allowable Parameter Combinations

The PLB_TEMAC is a synchronous pipe lined design. Due to the pipe lined architecture of receive and transmit operations and the interface with the packet FIFOs, the PLB Clock must be in the range of 42 MHz to 125 MHz to allow Ethernet operation at 1 Gb/s.

If Jumbo sized frames will be used in the system, the packet FIFOs must be deep enough to hold at least one complete frame.

Detailed Parameter Descriptions

C_DEV_BLK_ID

The block ID is reflected as a field in the Module Identification Registers (MIR). This may be used for identification and verification of correct register access capability.

C_PLB_CLK_PERIOD_PS

This clock period information is used by the IPIF sub-module for calculating the interrupt coalescing period. Interrupt coalescing is optional and is only used when using Scatter Gather DMA. The use of interrupt coalescing with Scatter Gather DMA can significantly improve Ethernet performance while reducing processor utilization.

C_FAMILY

The family parameter is required to implement the core using family specific architecture features. This parameter is automatically updated by the EDK tools based on the project target device information.

C_IPIF_RDFIFO_DEPTH and C_IPIF_WRFIFO_DEPTH

These parameters set the depth (in bits) of the packet FIFOs. The depth of the FIFOs impacts the number of BRAMS used in the system and only has a minor impact on logic resources used.

Like the C_MAC_FIFO_DEPTH parameter, a deeper packet FIFO will use more resources but offers the potential for higher bandwidth traffic on the Ethernet bus while reducing processor utilization.

The minimum FIFO size (16384 bits = 2048 bytes) is large enough to hold one maximum "normal" frame of 1518 bytes. However, increasing the packet FIFO to hold at least two frames can significantly increase performance. A minimum packet FIFO size of 131,072 bits = 16,384 bytes is required to hold one maximum size "jumbo" frame.

In cases where packets are primarily small it may not be useful to have a large packet FIFO. For example, if all of the packets are 64 bytes, a maximum depth (64 entry) length and status FIFO will limit queued packets to 4096 total bytes in the packet FIFO.

C_MAC_FIFO_DEPTH

This parameter is used to select a depth for the transmit status, transmit length, receive status, and receive length registers. These registers are actually FIFOs and the depth represents the maximum number of entries that may be queued up before an overflow condition occurs.

Selecting a larger depth will use more resources but offers the potential for higher bandwidth traffic on the Ethernet bus while reducing processor utilization. The number of transmit and receive packets you will be able to queue up will be limited by the number of entries in these FIFOs or by the size of the packet FIFOs whichever fills up first.

If the packet sizes will be small, these FIFOs will fill up more quickly and should be as large as possible and conversely, if the packet sizes are large, the packet FIFOs will fill up first and should be as large as possible. If the packet size is unknown or is distributed over all sizes than these FIFOs and the packet FIFOs should both be made as large as possible.

C_BASEADDR and C_HIGHADDR

These values are used to generate the read and write enables for the FIFOs and registers and are byte addresses. The address range defined by these parameters must be at least 0x4000. For example, if the C_BASEADDR is set to 0x10000000, then C_HIGHADDR must be set to at least 0x10003FFF. These parameters must be initialized since the default values have been selected so that they will generate an error during build if left unchanged.

C_INCLUDE_DEV_MIR

This parameter includes or excludes the Module Identification Registers in the IPIF sub module. This includes the device MIR, packet FIFO MIRs, and the DMA MIR if the DMA module is included in the design. This will reduce resource utilization in those cases where the MIR functionality is not required.

The software reset function (which shares an address with the Device MIR register) is not dependent on this parameter and may be present even when the MIR registers are not. However, the Device MIR register is dependent on the C_RESET_PRESENT parameter and will not be available if the software reset function is excluded from the design.

C_INCLUDE_DRE

This parameter includes or excludes the Data Re-Alignment Engine function in the IPIF submodule. This enhancement can be added to the front end of the PLB IPIF WrFIFO to improve DMA Tx throughput when source data is not aligned to 64-bit address boundaries. This scenario is encountered frequently when off the shelf operating systems such as VxWorks are employed in a PPC405 system.

The DRE has been developed to solve this problem. It is a hardware accelerator that takes arbitrarily aligned input data stream (64-bits wide, arranged in 8 byte lanes) and realigns it to an arbitrary byte alignment (64-bits wide, arranged in 8 byte lanes) at the output. This hardware accelerator removes the need for the intermediate buffer copy by the processor element.

Excluding this function will allow some reduction in resources.

C_INCLUDE_RESET

This parameter includes or excludes the software reset function. This may allow some reduction in resources if the software reset function is not required.

The software reset function (which shares an address with the Device MIR register) is not dependent on the C_DEV_MIR_ENABLE parameter and may be present even when the MIR registers are not. However, the Device MIR register is dependent on the C_RESET_PRESENT parameter and will not be available if the software reset function is excluded from the design.

C_INCLUDE_DEV_PENCODER

This parameter includes or excludes the Device Interrupt Identification Register. This may allow some reduction in resources if the Device Interrupt Identification Register is not required.

The Device Interrupt Identification Register is an ordinal value output of an interrupt priority encoder. The value indicates which interrupt source, if any, has a pending interrupt. A value of 0x80 indicates that there are no pending interrupts, otherwise, the value gives the bit position in the DIPR of the highest priority interrupt that is pending.

The priority is highest for the interrupt bit in the LSB position (bit 31), which reports as ID value 0x00, and decreases in priority (and increases in the reported ID value) for each successively more significant position (i.e. going left). Including this function may simplify software processing of interrupts in their order of importance.

C_DMA_TYPE

This parameter controls the inclusion of simple DMA (2), scatter gather DMA (3) or no DMA (1) functionality. Choosing to include either DMA also automatically includes an PLB master bus interface.

Including DMA can significantly increase resources but offers the potential for higher bandwidth traffic on the Ethernet bus while reducing processor overhead and increasing bandwidth on the processor bus.

For more information about DMA please refer to the Processor IP Reference Guide.

C_INCLUDE_DMA_INTR_COALESCE

This parameter includes or excludes the interrupt coalesce functionality. This may allow some reduction in resources if the interrupt coalesce function is not required.

Interrupt coalescing is only available when scatter gather DMA is being used and allows multiple interrupt events to queue up before being processed while guaranteeing a maximum wait bound. This can reduce processor overhead by reducing the number of interrupts processed.

For more information about interrupt coalescing and scatter gather DMA please refer to the Processor IP Reference Guide.

C_PLB_NUM_MASTERS

This parameter is automatically updated by the EDK tools based on the project information.

C_PLB_MID_WIDTH

This parameter is automatically updated by the EDK tools based on the project information.

C_OPB_AWIDTH and C_OPB_DWIDTH

These parameters should always be set to 32 and 64 respectively.

C_TEMAC_IS_SOFT

In future releases this core may support a soft version of the HARD_TEMAC for use in other FPGA devices. Currently only the HARD_TEMAC is supported and this parameter must always be set to ’0’.

C_TEMAC_DCR_HOST

In future releases this core will allow the user to select between a DCR bus interface or the PLB interface to the HARD_TEAMC registers. In this release the DCR bus interface is not supported so this parameter should be ’0’.

C_TEMAC_INST

This parameter indicates to which half of the HARD_TEMAC this PLB_TEMAC is connected. If the PLB_TEMAC is connected to the first half of the HARD_TEMAC (V4EMACSRC connected to V4EMACDST0) than this value should be ’0’ otherwise it is ’1’. If only one half of the HARD_TEAMC is used than the PLB_TEMAC must be connected to the first half and this value must be ’0’. This release only supports single PLB_TEMAC systems so this parameter should be ’0’.

C_TEMAC_BOTH_USED

This parameter indicates if the system contains one PLB_TEMAC connected to a HARD_TEMAC (value of ’0’) or if there are two PLB_TEMACs connected to the same HARD_TEMAC (value of ’1’). This release only supports a single PLB_TEMAC system for each HARD_TEMAC and so this parameter should be ’0’.

PLB_TEMAC I/O Signals

The PLB_TEMAC core uses the "transparent bus" format to simplify generation of embedded systems by greatly simplifying the connection of signals between the PLB_TEMAC and HARD_TEMAC cores. This is the same technique that allows the EDK tools to automatically connect the PLB signals.

The ports on the PLB_TEMAC which connect to the HARD_TEMAC are grouped into a virtual bus called V4EMACSRC. The corresponding signals on the HARD_TEMAC are grouped into two virtual busses called V4EMACDST0 and V4EMACDST1 depending on which half of the HARD_TEMAC the signals are used.

Most of what needs to be done to connect a PLB_TEMAC to 1/2 of a HARD_TEMAC is to designate which PLB_TEMAC connects to which half of the HARD_TEMAC. This is done by assigning the PLB_TEMAC virtual bus a name that matches the name assigned to the half of the HARD_TEMAC. An example of how this is done will be shown later in the document. In the signal list below, those signals that are assigned to the virtual bus are designated with an interface value of V4EMACSRC.

When only using half of the HARD_TEMAC, the PLB_TEMAC must be connected to the half of the HARD_TEMAC designated with the virtual bus V4EMACDST0. The half of the HARD_TEMAC designated with the virtual bus V4EMACDST1 is only connected to a PLB_TEMAC when using both halves of the HARD_TEMAC. When using only one PLB_TEMAC in a system the shared host signals are unused and should be left unconnected. The inputs will automatically be tied high or low as required.

When using both halves of the HARD_TEMAC, the two PLB_TEMACs must be connected together in order to share the one host interface connection to the HARD_TEMAC. The PLB_TEMAC connected to V4EMACDST0 will drive the host interface based on its own requests and those requests it receives from the PLB_TEMAC connected to V4EMACDST1. The signals used by the PLB_TEMAC to share the host interface are designated with an interface value of Shared Host in the signal list below.

The external I/O signals for the PLB_TEMAC are listed in Table 2.

Table 2: **PLB_TEMAC I/O Signals**

Grouping		Signal Name	Interface	I/O	Description
PLB System Signals	P1	PLB_Clk	System	I	System clock
	P2	PLB_Rst	System	I	System Reset (active high)
	P3	IP2INTC_Irpt	System	O	System Interrupt
PLB Arbiter Signals	P4	PLB_ABus(0:C_PLB_AWIDTH-1)	PLB bus	I	PLB address bus
	P5	PLB_BE(0:(C_PLB_DWIDTH/8)-1)	PLB bus	I	PLB byte enables
	P6	PLB_wrDBus(0:C_PLB_DWIDTH-1)	PLB bus	I	PLB write data bus
	P7	PLB_RNW	PLB bus	I	PLB Read not Write
	P6	PLB_PAValid	PLB bus	I	PLB primary address valid indicator
	P7	PLB_SAValid	PLB bus	I	PLB secondary address valid indicator
	P8	PLB_rdPrim	PLB bus	I	PLB secondary to primary read request indicator
	P9	PLB_wrPrim	PLB bus	I	PLB secondary to primary write request indicator
	P10	PLB_masterID(0:C_PLB_MID_WIDTH -1)	PLB bus	I	PLB current master indentifier
	P11	PLB_abort	PLB bus	I	PLB abort bus request indicator
	P12	PLB_buslock	PLB bus	I	PLB bus lock
	P13	PLB_MSize(0:1)	PLB bus	I	PLB master data bus size
	P14	PLB_size(0:3)	PLB bus	I	PLB transfer size
	P15	PLB_type(0:2)	PLB bus	I	PLB transfer type
	P16	PLB_compress	PLB bus	I	PLB compressed data transfer indicator
	P17	PLB_guarded	PLB bus	I	PLB guarded transfer indicator
	P18	PLB_ordered	PLB bus	I	PLB synchronize transfer indicator
	P19	PLB_lockErr	PLB bus	I	PLB lock error indicator
	P20	PLB_wrBurst	PLB bus	I	PLB burst write transfer indicator
	P21	PLB_rdBurst	PLB bus	I	PLB burst read transfer indicator
	P22	PLB_pendReq	PLB bus	I	PLB pending bus request indicator
	P23	PLB_pendPri(0:1)	PLB bus	I	PLB pending request priority
	P24	PLB_reqPri(0:1)	PLB bus	I	PLB current request priority
PLB Master Signals	P25	PLB_MSSize(0:1)	PLB bus	I	PLB slave data bus size
	P26	M_request	PLB bus	O	Master bus request
	P27	M_priority(0:1)	PLB bus	O	Master bus request priority

Table 2: **PLB_TEMAC I/O Signals *(Continued)***

Grouping		Signal Name	Interface	I/O	Description
	P28	M_buslock	PLB bus	O	Master bus lock
	P29	M_RNW	PLB bus	O	Master Read not Write
	P30	M_BE(0:(C_PLB_DWIDTH/8)-1)	PLB bus	O	Master byte enables
	P31	M_MSize(0:1)	PLB bus	O	Master data bus size
	P32	M_size(0:3)	PLB bus	O	Master transfer size
	P33	M_type(0:2)	PLB bus	O	Master transfer type
	P34	M_compress	PLB bus	O	Master compressed data transfer indicator
	P35	M_guarded	PLB bus	O	Master guarded transfer indicator
	P36	M_ordered	PLB bus	O	Master synchronize transfer indicator
	P37	M_lockErr	PLB bus	O	Master lock error indicator
	P38	M_abort	PLB bus	O	Master abort bus request indicator
	P39	M_abus(0:C_PLB_AWIDTH-1)	PLB bus	O	Master address bus
	P40	M_wrDBus(0:C_PLB_DWIDTH-1)	PLB bus	O	Master write data bus
	P41	M_wrBurst	PLB bus	O	Master burst write transfer indicator
	P42	M_rdBurst	PLB bus	O	Master burst read transfer indicator
	P43	PLB_MAddrAck	PLB bus	I	PLB master address acknowledge
	P44	PLB_MRearbitrate	PLB bus	I	PLB master bus rearbitrate indicator
	P45	PLB_MBusy	PLB bus	I	PLB master slave busy indicator
	P46	PLB_MErr	PLB bus	I	PLB master slave error indicator
	P47	PLB_MWrDAck	PLB bus	I	PLB master write data acknowledge
	P48	PLB_RdDBus(0:C_PLB_DWIDTH-1)	PLB bus	I	PLB master read data bus
	P49	PLB_MRdWdAddr(0:3)	PLB bus	I	PLB master read word address
	P50	PLB_MRdDAck	PLB bus	I	PLB master read data acknowledge
	P51	PLB_MRdBTerm	PLB bus	I	PLB master terminate read burst indicator
	P52	PLB_MWrBTerm	PLB bus	I	PLB master terminate write burst indicator
PLB Slave Signals	P53	Sl_rdDBus(0:C_PLB_DWIDTH-1)	PLB bus	O	Slave read data bus
	P54	Sl_addrAck	PLB bus	O	Slave address acknowledge
	P55	Sl_SSize(0:1)	PLB bus	O	Slave data bus size
	P56	Sl_wait	PLB bus	O	Slave wait indicator
	P57	Sl_rearbitrate	PLB bus	O	Slave rearbitrate bus indicator
	P58	Sl_wrDAck	PLB bus	O	Slave write data acknowledge

Table 2: **PLB_TEMAC I/O Signals *(Continued)***

Grouping		Signal Name	Interface	I/O	Description
	P59	Sl_wrComp	PLB bus	O	Slave write transfer complete indicator
	P60	Sl_wrBTerm	PLB bus	O	Slave terminate write burst transfer
	P61	Sl_rdWdAddr(0:3)	PLB bus	O	Slave read word address
	P62	Sl_rdDAck	PLB bus	O	Slave read data acknowledge
	P63	Sl_rdComp	PLB bus	O	Slave read transfer complete indicator
	P64	Sl_rdBTerm	PLB bus	O	Slave terminate read burst transfer
	P65	Sl_MBusy(0:C_PLB_NUM_MASTERS-1)	PLB bus	O	Slave busy indicator
	P66	Sl_MErr(0:C_PLB_NUM_MASTERS-1)	PLB bus	O	Slave error indicator
TEMAC Signals	P67	Emac_Reset	EMAC	O	This reset signal is a combination of the PLB bus reset and the software reset if included in the build. This signal may be used to reset the HARD_TEMAC.
	P68	TieEmacConfigVec(79:0)	V4EMACSRC	O	The values of these tie-off pins are loaded into the Ethernet MAC at power-up or when the Ethernet MAC is reset.
	P69	TieEmacUniCastAddr(47:0)	V4EMACSRC	O	These values are used to set an initial Ethernet MAC unicast address used by the address filter block to see if the incoming frame is destined for the Ethernet MAC. The address is ordered for the least significant byte in the register to have the first byte transmitted or received; for example, an Ethernet MAC address of 06-05-04-03-02-01 is stored in byte [47:0] as 0x010203040506.
	P70	EmacClientAnInterrupt	V4EMACSRC	I	EMAC client Auto negotiation interrupt
	P71	ClientEmacDcmLocked	V4EMACSRC	O	DCM stable indicator
	P72	ClientEmacPauseReq	V4EMACSRC	O	Pause Request
	P73	ClientEmacPauseVal(15:0)	V4EMACSRC	O	Pause Value
	P74	ClientEmacRxClientClkIn	V4EMACSRC	O	Receive Client Clock input
	P75	EmacClientRxBadFrame	V4EMACSRC	I	Receive frame is bad
	P76	EmacClientRxClientClkOut	V4EMACSRC	I	Receive Client Clock output
	P77	EmacClientRxd(15:0)	V4EMACSRC	I	Client Receive Data
	P78	EmacClientRxdVld	V4EMACSRC	I	Client receive data valid

Table 2: **PLB_TEMAC I/O Signals *(Continued)***

Grouping		Signal Name	Interface	I/O	Description
	P79	EmacClientRxdVldMsw	V4EMACSRC	I	Client receive data valid on most significant word
	P80	EmacClientRxdVreg6	V4EMACSRC	I	Receive data valid early registration
	P81	EmacClientRxFrameDrop	V4EMACSRC	I	Receive frame dropped indication
	P82	EmacClientRxGoodFrame	V4EMACSRC	I	Valid receive frame indicator
	P83	EmacClientRxStats(6:0)	V4EMACSRC	I	Receive statistics
	P84	EmacClientRxStatsByteVld	V4EMACSRC	I	Receive statistics byte valid
	P85	EmacClientRxStatsVld	V4EMACSRC	I	Receive statistics valid indicator
	P86	ClientEmacTxClientClkIn	V4EMACSRC	O	Transmit clock input
	P87	ClientEmacTxd(15:0)	V4EMACSRC	O	Transmit data
	P88	ClientEmacTxdVld	V4EMACSRC	O	Transmit data valid
	P89	ClientEmacTxdVldMsw	V4EMACSRC	O	Transmit data valid most significant word
	P90	ClientEmacTxFirstByte	V4EMACSRC	O	Transmit first byte indicator
	P91	ClientEmacTxGmiiMiiClkIn	V4EMACSRC	O	Transmit clock for MII GMII
	P92	ClientEmacTxIfgDelay(7:0)	V4EMACSRC	O	Interframe gap delay for tx.
	P93	ClientEmacTxUnderRun	V4EMACSRC	O	Tx under run
	P94	EmacClientTxAck	V4EMACSRC	I	Transmit acknowlege
	P95	EmacClientTxClientClkOut	V4EMACSRC	I	Transmit clock from EMAC
	P96	EmacClientTxCollision	V4EMACSRC	I	Transmit collision indicator
	P97	EmacClientTxGmiiMiiClkOut	V4EMACSRC	I	Transmit clock for MII GMII
	P98	EmacClientTxRetransmit	V4EMACSRC	I	Retransmit indication
	P99	EmacClientTxStats	V4EMACSRC	I	Transmit statistics
	P100	EmacClientTxStatsByteVld	V4EMACSRC	I	Transmit statistics byte valid
	P101	EmacClientTxStatsVld	V4EMACSRC	I	Transmit statistics valid
	P102	HostMiimRdy	V4EMACSRC	I	MII management / host IF is ready
	P103	HostRdData(31:0)	V4EMACSRC	I	Data read through Host IF
	P104	HostAddr(9:0)	V4EMACSRC	O	Host address
	P105	Dcremacenable	V4EMACSRC	O	Enables the DCR bus host interface access. When this signal is deasserted, the generic host interface is selected. This signal is tied to the parameter C_TEMAC_DCR_HOST. **(Only '0' is supported in this release)**
	P106	HostClk	V4EMACSRC	O	Host clock (1 - 100 MHz)
	P107	HostEmac1Sel	V4EMACSRC	O	Host select for EMAC no. 1
	P108	HostMiimSel	V4EMACSRC	O	Host select for MII management

Table 2: **PLB_TEMAC I/O Signals *(Continued)***

Grouping		Signal Name	Interface	I/O	Description
	P109	HostOpCode(1:0)	V4EMACSRC	O	Host read / write indication
	P110	HostReq	V4EMACSRC	O	Host IF request
	P111	HostWrData(31:0)	V4EMACSRC	O	Data written through Host IF
Parasitic Host	P112	Prv_Host_Req	Shared Host	I	Provider Host port request
	P113	Prv_Host_Addr(22:31)	Shared Host	I	Provider Host port address
	P114	Prv_Host_Rd_Req	Shared Host	I	Provider Host port read request
	P115	Prv_Host_Rd_Ack	Shared Host	O	Provider Host port read acknowledge
	P116	Prv_Host_Rd_Data(0:31)	Shared Host	O	Provider Host port read data
	P117	Prv_Host_Wr_Req	Shared Host	I	Provider Host port write request
	P118	Prv_Host_Wr_Ack	Shared Host	O	Provider Host port write acknowlege
	P119	Prv_Host_Wr_Data(0:31)	Shared Host	I	Provider Host port write data
	P120	Cmd_Host_Req	Shared Host	O	Command Host port request
	P121	Cmd_Host_Addr(22:31)	Shared Host	O	Command Host port address
	P122	Cmd_Host_Rd_Req	Shared Host	O	Command Host port read request
	P123	Cmd_Host_Rd_Ack	Shared Host	I	Command Host port read acknowlege
	P124	Cmd_Host_Rd_Data(0:31)	Shared Host	I	Command Host port read data
	P125	Cmd_Host_Wr_Req	Shared Host	O	Command Host port write request
	P126	Cmd_Host_Wr_Ack	Shared Host	I	Command Host port write acknowlege
	P127	Cmd_Host_Wr_Data(0:31)	Shared Host	O	Command Host port write data

Figure 4 shows a PLB_TEMAC system with one PLB_TEMAC while Figure 5 shows a dual PLB_TEMAC system connected to the HARD_TEMAC.

Figure 6 is an example of part of an EDK .mhs file showing a single PLB_TEMAC system using the GMII PHY interface.

Figure 7 and Figure 8 are examples of part of an EDK .mhs file showing a dual PLB_TEMAC system using the GMII PHY interface. This example shows the connections between the two PLB_TEMACs for the shared host interface to the HARD_TEMAC as well as the use of the "BUS_INTERFACE" keywords to specify which PLB_TEMAC connects to which half of the HARD_TEMAC.

Note the values used for the PLB_TEMAC parameters C_TEMAC_INST and C_TEMAC_BOTH_USED to differentiate this system from the single PLB_TEMAC system.

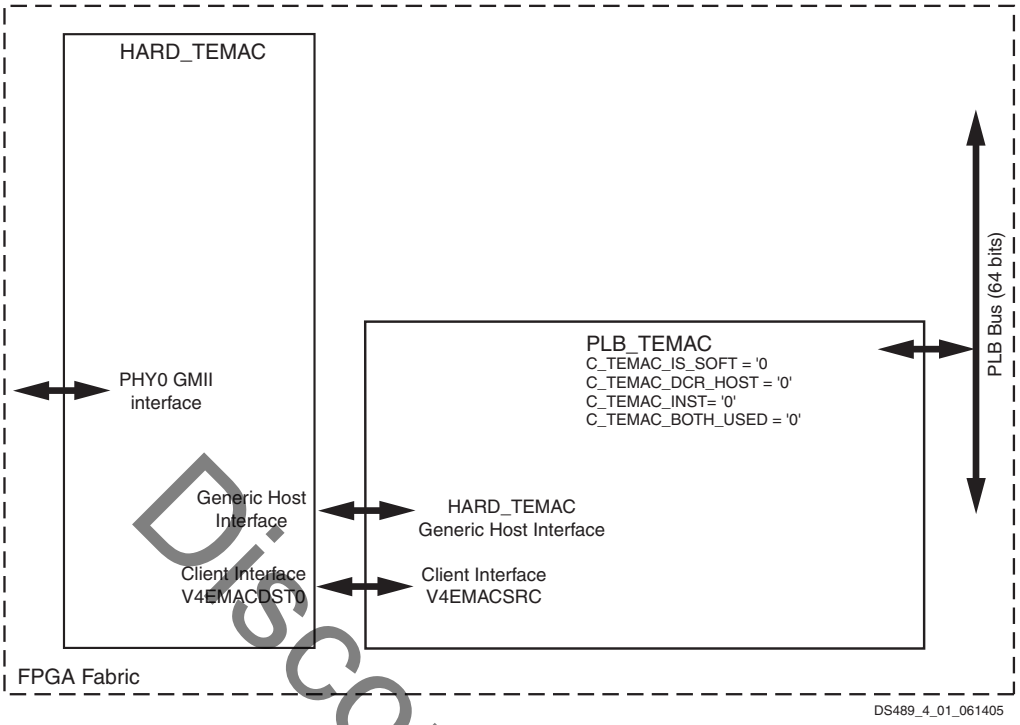

Figure 4: **System with one PLB_TEMAC**

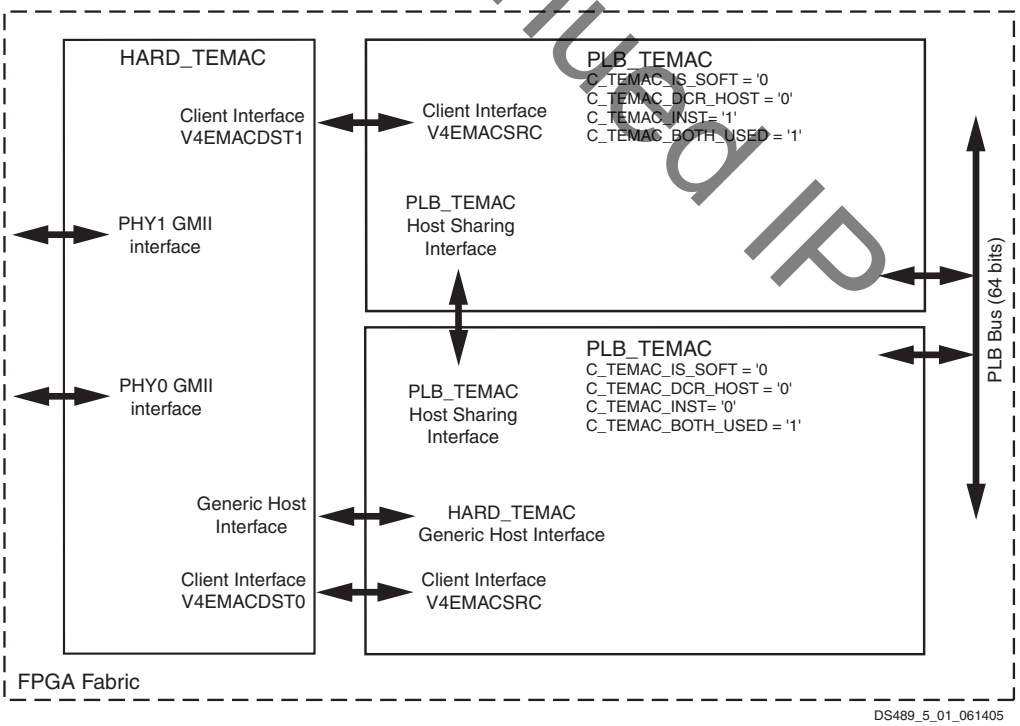

Figure 5: **System with two PLB_TEMACs**

```
BEGIN plb_temac
 PARAMETER INSTANCE = plb_temac_0
 PARAMETER HW_VER = 2.00.a
 PARAMETER C_PLB_CLK_PERIOD_PS = 10000
 PARAMETER C_BASEADDR = 0x80400000
 PARAMETER C_HIGHADDR = 0x8040ffff
 PARAMETER C_TEMAC_IS_SOFT = 0
 PARAMETER C_TEMAC_INST = 0
 PARAMETER C_TEMAC_BOTH_USED = 0
 PARAMETER C_TEMAC_DCR_HOST = 0
 BUS_INTERFACE MSPLB = plb
 BUS_INTERFACE V4EMACSRC = V4MEMAC_PLBINTFCE
 PORT PLB_Clk = sys_clk_s
 PORT IP2INTC_Irpt = Ethernet_MAC_IP2INTC_Irpt
 PORT Emac_Reset = Emac0_Reset
 PORT EmacClientTxGmiimiiClkOut = emac_client_tx_gmii_mii_clk_out
END

BEGIN hard_temac
 PARAMETER INSTANCE = hard_temac_0
 PARAMETER HW_VER = 1.00.a
 BUS_INTERFACE V4EMACDST0 = V4MEMAC_PLBINTFCE
 PORT RESET = Emac0_Reset
 PORT PHYEMAC0GTXCLK = PHYEMAC0GTXCLK
 PORT EMAC0PHYTXCLK = Emac0PhyTxClk
 PORT EMAC0PHYTXD = Emac0PhyTxD
 PORT EMAC0PHYTXEN = Emac0PhyTxEn
 PORT EMAC0PHYTXER = Emac0PhyTxEr
 PORT PHYEMAC0RXCLK = PhyEmac0RxClk
 PORT PHYEMAC0CRS = PhyEmac0Crs
 PORT PHYEMAC0COL = PhyEmac0Col
 PORT PHYEMAC0RXD = PhyEmac0RxD
 PORT PHYEMAC0RXDV = PhyEmac0RxDv
 PORT PHYEMAC0RXER = PhyEmac0RxEr
 PORT EMAC0PHYMCLKOUT = Emac0PhyMClkOut
 PORT EMAC0PHYMD = Emac0PhyMD
 PORT PHYEMAC0MIITXCLK = PhyEmac0MiiTxClk
 PORT DCRHOSTDONEIR = DCRHOSTDONEIR
 PORT EMAC0CLIENTTXGMIIMIICLKOUT = emac_client_tx_gmii_mii_clk_out
END
```

DS489_6_01_061405

Figure 6: **EDK .mhs file segment for a single PLB_TEMAC system using the GMII interface**

```
BEGIN plb_temac
 PARAMETER INSTANCE = plb_temac_0
 PARAMETER HW_VER = 2.00.a
 PARAMETER C_PLB_CLK_PERIOD_PS = 10000
 PARAMETER C_BASEADDR = 0x80400000
 PARAMETER C_HIGHADDR = 0x8040ffff
 PARAMETER C_TEMAC_INST = 0
 PARAMETER C_TEMAC_BOTH_USED = 1
 PARAMETER C_TEMAC_DCR_HOST = 0
 BUS_INTERFACE MSPLB = plb
 BUS_INTERFACE V4EMACSRC = emacPort0
 PORT PLB_Clk = sys_clk_s
 PORT IP2INTC_Irpt = Ethernet_MAC0_IP2INTC_Irpt
 PORT Emac_Reset = Emac0_Reset
 PORT EmacClientTxGmiimiiClkOut = emacPort0_emac_client_tx_gmii_mii_clk_out
 PORT Prv_Host_Req = Prv_Host_Req
 PORT Prv_Host_Addr = Prv_Host_Addr
 PORT Prv_Host_Rd_Req = Prv_Host_Rd_Req
 PORT Prv_Host_Rd_Ack = Prv_Host_Rd_Ack
 PORT Prv_Host_Rd_Data = Prv_Host_Rd_Data
 PORT Prv_Host_Wr_Req = Prv_Host_Wr_Req
 PORT Prv_Host_Wr_Ack = Prv_Host_Wr_Ack
 PORT Prv_Host_Wr_Data = Prv_Host_Wr_Data
 PORT Cmd_Host_Req = Cmd_Host_Req
 PORT Cmd_Host_Addr = Cmd_Host_Addr
 PORT Cmd_Host_Rd_Req = Cmd_Host_Rd_Req
 PORT Cmd_Host_Rd_Ack = Cmd_Host_Rd_Ack
 PORT Cmd_Host_Rd_Data = Cmd_Host_Rd_Data
 PORT Cmd_Host_Wr_Req = Cmd_Host_Wr_Req
 PORT Cmd_Host_Wr_Ack = Cmd_Host_Wr_Ack
 PORT Cmd_Host_Wr_Data = Cmd_Host_Wr_Data
END

BEGIN plb_temac
 PARAMETER INSTANCE = plb_temac_1
 PARAMETER HW_VER = 2.00.a
 PARAMETER C_PLB_CLK_PERIOD_PS = 10000
 PARAMETER C_BASEADDR = 0x80500000
 PARAMETER C_HIGHADDR = 0x8050ffff
 PARAMETER C_TEMAC_IS_SOFT = 0
 PARAMETER C_TEMAC_INST = 1
 PARAMETER C_TEMAC_BOTH_USED = 1
 PARAMETER C_TEMAC_DCR_HOST = 0
 BUS_INTERFACE MSPLB = plb
 BUS_INTERFACE V4EMACSRC = emacPort1
 PORT PLB_Clk = sys_clk_s
 PORT IP2INTC_Irpt = Ethernet_MAC1_IP2INTC_Irpt
 PORT Emac_Reset = Emac1_Reset
 PORT EmacClientTxGmiimiiClkOut = emacPort1_emac_client_tx_gmii_mii_clk_out
 PORT Prv_Host_Req = Cmd_Host_Req
 PORT Prv_Host_Addr = Cmd_Host_Addr
 PORT Prv_Host_Rd_Req = Cmd_Host_Rd_Req
 PORT Prv_Host_Rd_Ack = Cmd_Host_Rd_Ack
 PORT Prv_Host_Rd_Data = Cmd_Host_Rd_Data
 PORT Prv_Host_Wr_Req = Cmd_Host_Wr_Req
 PORT Prv_Host_Wr_Ack = Cmd_Host_Wr_Ack
 PORT Prv_Host_Wr_Data = Cmd_Host_Wr_Data
 PORT Cmd_Host_Req = Prv_Host_Req
 PORT Cmd_Host_Addr = Prv_Host_Addr
 PORT Cmd_Host_Rd_Req = Prv_Host_Rd_Req
 PORT Cmd_Host_Rd_Ack = Prv_Host_Rd_Ack
 PORT Cmd_Host_Rd_Data = Prv_Host_Rd_Data
 PORT Cmd_Host_Wr_Req = Prv_Host_Wr_Req
 PORT Cmd_Host_Wr_Ack = Prv_Host_Wr_Ack
 PORT Cmd_Host_Wr_Data = Prv_Host_Wr_Data
END
```

DS489_7_01_061405

Figure 7: **EDK .mhs file segment for a dual PLB_TEMAC system showing shared host interface**

```
BEGIN hard_temac
 PARAMETER INSTANCE = hard_temac_0
 PARAMETER HW_VER = 1.00.a
 BUS_INTERFACE V4EMACDST0 = emacPort0
 BUS_INTERFACE V4EMACDST1 = emacPort1
 PORT RESET = Emac0_Reset
 PORT PHYEMAC0GTXCLK = PHYEMAC0GTXCLK
 PORT EMAC0PHYTXCLK = Emac0PhyTxClk
 PORT EMAC0PHYTXD = Emac0PhyTxD
 PORT EMAC0PHYTXEN = Emac0PhyTxEn
 PORT EMAC0PHYTXER = Emac0PhyTxEr
 PORT PHYEMAC0RXCLK = PhyEmac0RxClk
 PORT PHYEMAC0CRS = PhyEmac0Crs
 PORT PHYEMAC0COL = PhyEmac0Col
 PORT PHYEMAC0RXD = PhyEmac0RxD
 PORT PHYEMAC0RXDV = PhyEmac0RxDv
 PORT PHYEMAC0RXER = PhyEmac0RxEr
 PORT EMAC0PHYMCLKOUT = Emac0PhyMClkOut
 PORT EMAC0PHYMD = Emac0PhyMD
 PORT PHYEMAC0MIITXCLK = PhyEmac0MiiTxClk
 PORT EMAC0CLIENTTXGMIIMIICLKOUT = emacPort0_emac_client_tx_gmii_mii_clk_out
 PORT PHYEMAC1GTXCLK = PHYEMAC0GTXCLK
 PORT EMAC1PHYTXCLK = Emac1PhyTxClk
 PORT EMAC1PHYTXD = Emac1PhyTxD
 PORT EMAC1PHYTXEN = Emac1PhyTxEn
 PORT EMAC1PHYTXER = Emac1PhyTxEr
 PORT PHYEMAC1RXCLK = PhyEmac1RxClk
 PORT PHYEMAC1CRS = PhyEmac1Crs
 PORT PHYEMAC1COL = PhyEmac1Col
 PORT PHYEMAC1RXD = PhyEmac1RxD
 PORT PHYEMAC1RXDV = PhyEmac1RxDv
 PORT PHYEMAC1RXER = PhyEmac1RxEr
 PORT EMAC1PHYMCLKOUT = Emac1PhyMClkOut
 PORT EMAC1PHYMD = Emac1PhyMD
 PORT PHYEMAC1MIITXCLK = PhyEmac1MiiTxClk
 PORT EMAC1CLIENTTXGMIIMIICLKOUT = emacPort1_emac_client_tx_gmii_mii_clk_out
 PORT DCRHOSTDONEIR = DCRHOSTDONEIR
END
```

DS489_7a_01_061405

Figure 8: **EDK .mhs file segment for a dual PLB_TEMAC system using the GMII interface**

PLB_TEMAC Port Dependencies

The width of some of the PLB_TEMAC signals depend on parameters selected in the design. The dependencies between the PLB_TEMAC design parameters and I/O signals are shown in Table 3.

Table 3: **PLB_TEMAC Parameter Port Dependencies**

		Name	Affects	Depends	Relationship Description
Design Parameters	G18	C_PLB_DWIDTH	P5,P6, P30,P40, P48,P53		Specifies the Data Bus width
	G17	C_PLB_AWIDTH	P4,P39		Specifies the Address Bus width
	G15	C_PLB_NUM_MASTERS	P65,P66		Specifies the number of masters on the PLB bus
	G16	C_PLB_MID_WIDTH	P10		Specifies the Master ID bus width
	G13	C_DMA_TYPE	G14		Specifies if DMA is present and which type
	G14	C_DMA_INTR_COASLESCE		G13	Not used if scatter gather DMA not present (G13 is 2 or 1)

Table 3: **PLB_TEMAC Parameter Port Dependencies *(Continued)***

		Name	Affects	Depends	Relationship Description
I/O Signals	P5	PLB_BE(0:(C_PLB_DWIDTH/8)-1)		G18	Width varies with the size of the Data bus.
	P4	PLB_ABus(0:C_PLB_AWIDTH-1)		G17	Width varies with the size of the Address bus.
	P6	PLB_wrDBus(0:C_PLB_DWIDTH-1)		G18	Width varies with the size of the Data bus.
	P30	M_BE(0:(C_PLB_AWIDTH/8)-1)		G17	Width varies with the size of the Data bus.
	P39	M_ABus(0:C_PLB_AWIDTH-1)		G17	Width varies with the size of the Address bus.
	P40	M_wrDBus(0:C_PLB_DWIDTH-1)		G18	Width varies with the size of the Data bus.
	P48	PLB_RdDBus(0:C_PLB_DWIDTH-1)		G18	Width varies with the size of the Data bus.
	P53	SI_RdDBus(0:C_PLB_DWIDTH-1)		G18	Width varies with the size of the Data bus.
	P65	SI_MBusy(0:C_PLB_NUM_MASTERS-1)		G15	Width varies with the number of masters on the PLB bus
	P66	SI_MErr(0:C_PLB_NUM_MASTERS-1)		G15	Width varies with the number of masters on the PLB bus
	P10	PLB_masterID(0:C_PLB_MID_WIDTH-1)		G16	Width varies with the number of masters on the PLB bus

PLB_TEMAC Interrupt Interface

The interrupt signals generated by the PLB_TEMAC are managed by the Interrupt Source Controller in the PLB_TEMAC IPIF module. This interface provides many of the features commonly provided for interrupt handling.

Please refer to the Processor IP Reference Guide for more details. An overview diagram of the interrupt control structure is show in Figure 9.

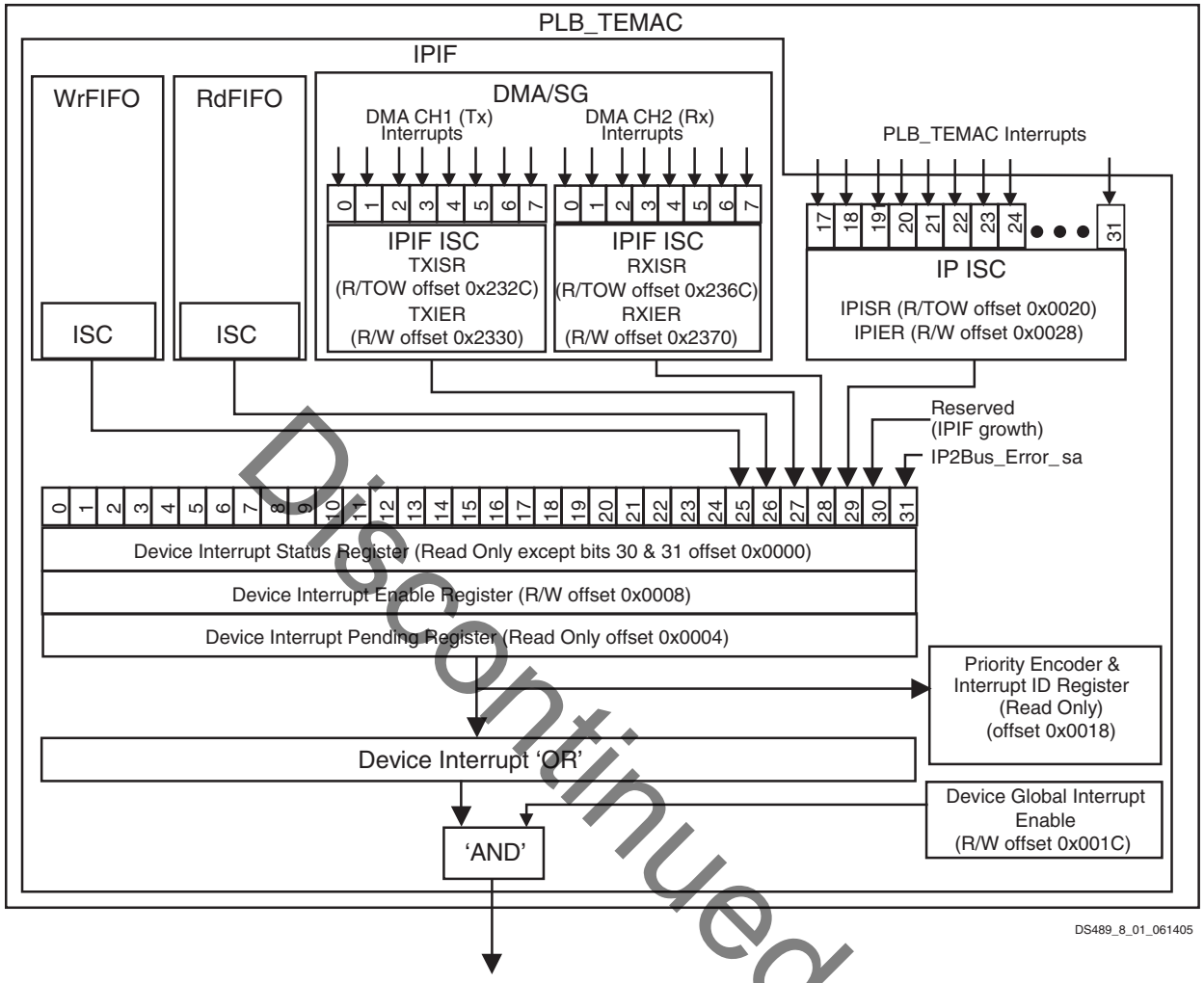

Figure 9: **Interrupt Control Structure**

The PLB_TEMAC interrupt values are available by reading the IP Interrupt Status Register (offset 0x0020); The interrupt descriptions are explained below.

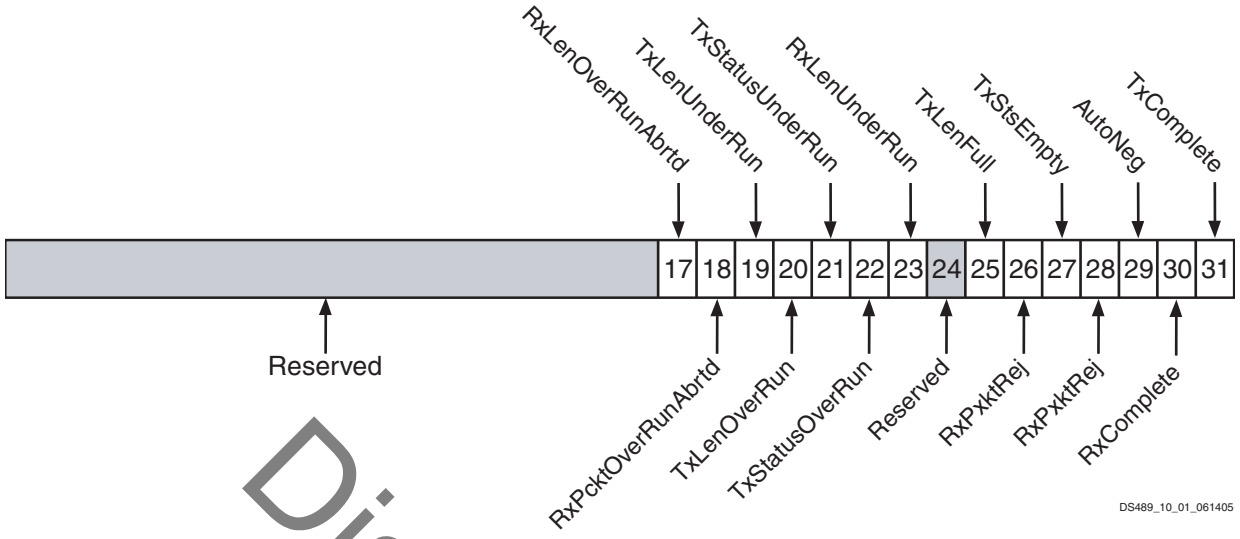

Figure 10: **IP Interrupt Status Register (offset 0x0020)**

Interrupt (data bus bit 31) -- Transmit complete interrupt

Indicates that at least one transmit has completed and that the transmit status word is available. This interrupt reflects the logic value of the transmit status FIFO not-empty flag. Consequently this interrupt will remain asserted until all values have been read from the transmit status FIFO and may “re-interrupt” after clearing the IPISR if data remains in the transmit status FIFO.

Interrupt (data bus bit 30) -- Receive complete interrupt

Indicates that at least one successful receive has completed and that the receive status word frame data and frame data length is available. This signal is not set for unsuccessful receives. This interrupt reflects the logic value of the receive status FIFO not-empty flag. Consequently this interrupt will remain asserted until all values have been read from the receive status FIFO and may “re-interrupt” after clearing the IPISR if data remains in the receive status FIFO.

Interrupt (data bus bit 29) -- Auto Negotiation Interrupt

Indicates the completion of auto negotiation in the HARD_TEMAC. Auto negotiation between the TEMAC and its link partner determines the communication attributes common to both. After auto negotiation the MII Management interface register should be queried for the negotiated status however the MII Management interface is not implemented in this release. Therefor there is no way to access the results of the auto negotiation. If possible auto negotiation should be disabled. Since the software must write the link speed and duplex to a HARD_TEMAC configuration register, the operation mode of the link must be forced to a known value. Future releases will include the MII Management interface and will allow the use of auto negotiation.

Interrupt (data bus bit 28) -- Receive Frame Rejected

Indicates that a received frame has been discarded due to addressing or transmission errors or because the reception would have overflowed either the receive packet FIFO or the Receive Packet Length or Status Registers.

Interrupt (data bus bit 27) -- Transmit Status FIFO Empty interrupt

This reflects the status of the transmit status FIFO empty flag. It may be used to indicate that the status words for all completed transmissions have been processed. Any other transmit frames already provided to the PLB_TEMAC are either queued for transmit or are currently being transmitted but have not yet completed. This active high signal remains active as long as the condition persists.

Interrupt (data bus bit 26) --Receive Length FIFO Empty interrupt

This reflects the status of the receive length FIFO empty flag. It may be used to indicate that the frame lengths for all successfully completed receives have been processed. The status of this FIFO should always track the status of the receive status FIFO. This active high signal remains active as long as the condition persists.

Interrupt (data bus bit 25) -- Transmit Length FIFO Full interrupt

This reflects the status of the transmit length FIFO full flag. It may be used to pause queueing of transmit frames until some of the queued frames have been processed by the PLB_TEMAC. This active high signal remains active as long as the condition persists.

Interrupt (data bus bit 24) -- Reserved

Interrupt (data bus bit 23) -- Receive Length FIFO Underrun interrupt

Indicates that an attempt was made to read the receive length FIFO when it was empty and that the data received is not valid. Once set, this bit can only be cleared with a reset.

Interrupt (data bus bit 22) -- Transmit Status FIFO Overrun interrupt

Indicates that the Transmit status FIFO became full following the transmission of a frame and data was lost. Care must be taken under these conditions to ensure that the transmit status words do not become out of sync with the originating frame information. To insure that more data is not lost, transmit status words stored in the FIFO should be processed to free up more locations. Once set, this bit can only be cleared with a reset.

Interrupt (data bus bit 21) -- Transmit Status FIFO Underrun interrupt

Indicates that an attempt was made to read the transmit status FIFO when it was empty and that the data received is not valid. Once set, this bit can only be cleared with a reset.

Interrupt (data bus bit 20) -- Transmit Length FIFO Overrun interrupt

Indicates that more transmit frames were written to the PLB_TEMAC transmit queue than the transmit length FIFO could store and data was lost. This is non-recoverable condition since some or all of the frame data may have been stored in the transmit data FIFO and it can not be removed.

Since there is not a transmit length entry for that frame, the transmit length and data FIFOs are no longer synchronized. This condition should be corrected by forcing an immediate reset of the transmit data FIFO and the transmit path in the PLB_TEMAC. Once set, this bit can only be cleared with a reset.

Interrupt (data bus bit 19) -- Transmit Length FIFO Underrun interrupt

Indicates that the PLB_TEMAC attempted to remove an entry from the transmit length FIFO following the completion of a transmission and there were no entries in the FIFO. This should never be possible and represents a serious error. This condition should be corrected by forcing an immediate reset of the transmit data FIFO and the transmit path in the PLB_TEMAC. condition. Once set, this bit can only be cleared with a reset.

Interrupt (data bus bit 18) -- Receive Frame FIFO Overrun interrupt

Indicates that at least one frame was dropped to prevent an overrun. This provides a graceful recovery for overrun conditions. Frames may be lost but an IP reset is not required. Received frames must be processed immediately to recover.

Interrupt (data bus bit 17) -- Receive Length FIFO Overrun Aborted

Indicates that at least one frame was dropped to prevent an overrun. This provides a graceful recovery for overrun conditions. Frames may be lost but an IP reset is not required. Received frames must be processed immediately to recover.

Interrupt (data bus bit 16 through 0) -- Reserved

PLB_TEMAC Registers Definition

The PLB_TEMAC provides access to registers in PLB_TEMAC, the PLB_TEMAC IPIF, and in the HARD_TEMAC core (connected via the host interface). All of the registers in the PLB_TEMAC are 32 bits wide or less. Some of the HARD_TEMAC

registers are more than 32 bits but can only be read 32 bits at a time. As a result, 32 bit addressing will be used for register accesses.

PLB_TEMAC IPIF Registers

The registers in Table 4 are contained in the PLB_TEMAC IPIF and are included for completeness of this specification. Detailed descriptions of these registers are provided in the IPIF specifications listed in *Reference Documents*.

Table 4: **IPIF Registers**

Register Name	PLB ADDRESS	Access
Device Interrupt Status Register	C_BASEADDR + 0x0000	Read
Device Interrupt Pending Register	C_BASEADDR + 0x0004	Read
Device Interrupt Enable Register	C_BASEADDR + 0x0008	Read/Write
Device Interrupt Identification Register	C_BASEADDR + 0x0018	Read
Device Global Interrupt Enable	C_BASEADDR + 0x001C	Read/Write
IP Interrupt Status Register	C_BASEADDR + 0x0020	Read/Toggle on Write
IP Interrupt Enable Register	C_BASEADDR + 0x0028	Read/Write
Device Software Reset (write) Module Identification (read) Register	C_BASEADDR + 0x0040	Read/Write
Write Packet FIFO reset (write) Module Identification (read)	C_BASEADDR + 0x2000	Read/Write
Write Packet FIFO Vacancy	C_BASEADDR + 0x2004	Read
Write Packet FIFO 64 bit wide data write port	C_BASEADDR + 0x2100	Write
Read Packet FIFO reset (write) Module Identification (read)	C_BASEADDR + 0x2010	Read/Write
Read Packet FIFO Occupancy	C_BASEADDR + 0x2014	Read
Read Packet FIFO 64 bit wide data read port	C_BASEADDR + 0x2200	Read
Transmit DMA & Scatter Gather Reset Register	C_BASEADDR + 0x2300	Write
Transmit DMA & Scatter Gather Module Identification Register	C_BASEADDR + 0x2300	Read
Transmit DMA & Scatter Gather Control Register	C_BASEADDR + 0x2304	Read/Write
Transmit DMA & Scatter Gather source address	C_BASEADDR + 0x2308	Read/Write
Transmit DMA & Scatter Gather destination address	C_BASEADDR + 0x230C	Read/Write
Transmit DMA & Scatter Gather start/length	C_BASEADDR + 0x2310	Read/Write
Transmit DMA & Scatter Gather Status Register	C_BASEADDR + 0x2314	Read
Transmit DMA & Scatter Gather Buffer Descriptor Address	C_BASEADDR + 0x2318	Read/Write
Transmit DMA Software Control Register	C_BASEADDR + 0x231C	Read/Write
Transmit DMA & Scatter Gather Unserviced Packet Count	C_BASEADDR + 0x2320	Read/Write
Transmit DMA & Scatter Gather Packet Count Threshold	C_BASEADDR + 0x2324	Read/Write
Transmit DMA & Scatter Gather Packet Wait Bound	C_BASEADDR + 0x2328	Read/Write
Transmit DMA & Scatter Gather Interrupt Status Register	C_BASEADDR + 0x232C	Read/toggle on Write
Transmit DMA & Scatter Gather Interrupt Enable Register	C_BASEADDR + 0x2330	Read/Write
Receive DMA & Scatter Gather Reset Register	C_BASEADDR + 0x2340	Write
Receive DMA & Scatter Gather Module Identification Register	C_BASEADDR + 0x2340	Read
Receive DMA & Scatter Gather Control Register	C_BASEADDR + 0x2344	Read/Write

Table 4: **IPIF Registers *(Continued)***

Register Name	PLB ADDRESS	Access
Receive DMA & Scatter Gather source address	C_BASEADDR + 0x2348	Read/Write
Receive DMA & Scatter Gather destination address	C_BASEADDR + 0x234C	Read/Write
Receive DMA & Scatter Gather start/length	C_BASEADDR + 0x2350	Read/Write
Receive DMA & Scatter Gather Status Register	C_BASEADDR + 0x2354	Read
Receive DMA & Scatter Gather Buffer Descriptor Address	C_BASEADDR + 0x2358	Read/Write
Receive DMA Software Control Register	C_BASEADDR + 0x235C	Read/Write
Receive DMA & Scatter Gather Unserviced Packet Count	C_BASEADDR + 0x2360	Read/Write
Receive DMA & Scatter Gather Packet Count Threshold	C_BASEADDR + 0x2364	Read/Write
Receive DMA & Scatter Gather Packet Wait Bound	C_BASEADDR + 0x2368	Read/Write
Receive DMA & Scatter Gather Interrupt Status Register	C_BASEADDR + 0x236C	Read/toggle on Write
Receive DMA & Scatter Gather Interrupt Enable Register	C_BASEADDR + 0x2370	Read/Write

PLB_TEMAC Registers

The registers in Table 5 are contained in the PLB_TEMAC core and are described in detail in this specification. The addresses for all registers are based on a parameter which is the base address for the entire PLB_TEMAC core. The address of each register is then calculated by an offset to the base address.

Table 5: **PLB_TEMAC Core Registers**

Register Name	PLB ADDRESS	Access
PLB_TEMAC Control Register (CR)	C_BASEADDR + 0x1000	Read/Write
Transmit Packet Length Register (TPLR)	C_BASEADDR + 0x1004	Read/Write
Transmit Status Register (TSR)	C_BASEADDR + 0x1008	Read
Receive Packet Length Register (RPLR)	C_BASEADDR + 0x100C	Read
Receive Status Register (RSR)	C_BASEADDR + 0x1010	Read
Interframe Gap Register (IFGP)	C_BASEADDR + 0x1014	Read/Write
Transmit Pause Packet Register (TPPR)	C_BASEADDR + 0x1018	Read/Write
MII Management Control Register (MGTCR)	C_BASEADDR + 0x101C	Read/Write
MII Management Data Register (MGTDR)	C_BASEADDR + 0x1020	Read/Write

HARD_TEMAC Core Registers

The HARD_TEMAC core registers are listed in Table 6. This description is appropriate for instances of PLB_TEMAC that use the PLB_IPIF to interface to the registers within the HARD_TEMAC core. Alternatively, these registers may be accessed through the DCR interface built into the processor block although **the DCR bus access in not supported in this release**.

The HARD_TEMAC has seven configuration registers (RXC0, RXC1, TXC, FCC, EMCFG, GMIC, and MC). These registers are accessed through the host interface and can be written to at any time. Both the receiver and transmitter logic only respond to configuration changes during IFGs. The configurable resets are the only exception, since the reset is immediate.

Address Filter Register access includes the address filter registers and the multicast address table registers. The HARD_TEMAC has five address filter registers (UAW0, UAW1, MAW0, MAW1, and AFM) with access through the host interface.

Some of the reset values of the HARD_TEMAC registers are determined by input signals to the HARD_TEMAC from the PLB_TEMAC. Where this is the case, the signals will be identified with the initial value provided by the PLB_TEMAC.

Table 6: **EMAC Core Registers**

Register Name	PLB ADDRESS	Access
Receive Configuration Word 0 (RXC0)	C_BASEADDR + 0x3200	Read/Write
Receive Configuration Word 1 (RXC1)	C_BASEADDR + 0x3240	Read/Write
Transmit Configuration (TXC)	C_BASEADDR + 0x3280	Read/Write
Flow Control Configuration (FCC)	C_BASEADDR + 0x32C0	Read/Write
EMAC Mode Configuration (EMCFG)	C_BASEADDR + 0x3300	Read (29:0) Read/Write (31:30)
RGMII / SGMII Configuration (GMIC)	C_BASEADDR + 0x3320	Read
Management Configuration (MC)	C_BASEADDR + 0x3340	Read/Write
Unicast Address Word 0 (UAW0)	C_BASEADDR + 0x3380	Read/Write
Unicast Address Word 1 (UAW1)	C_BASEADDR + 0x3384	Read/Write
Multicast Address Word 0 (MAW0)	C_BASEADDR + 0x3388	Read/Write
Multicast Address Word 1 (MAW1)	C_BASEADDR + 0x338C	Read/Write
Address Filter Mode (AFM)	C_BASEADDR + 0x3390	Read/Write

PLB_TEMAC Control Register (CR)

The PLB_TEMAC Control Register provides control of a few high level modes of operation of a PLB_TEMAC system. Broadcast and Multicast receive frames can be enabled or disabled. Additional receive Address filtering is performed in the HARD_TEMAC core under control of the HARD_TEMAC registers (please refer to the descriptions of those registers later in this section).

The duplex bit must be set to correspond to the half or full duplex operation of the system. If the PLB_TEMAC system is using auto negotiation, this would normally be read from the auto negotiation result register in the PHY device via the MII Management interface. Since the MII interface is not implemented in this release, the value of this bit and the duplex operation mode of the PLB_TEMAC system must be determined in another way. **This release of the PLB_TEMAC only implements FULL DUPLEX operation so this bit must always be set to '0'**.

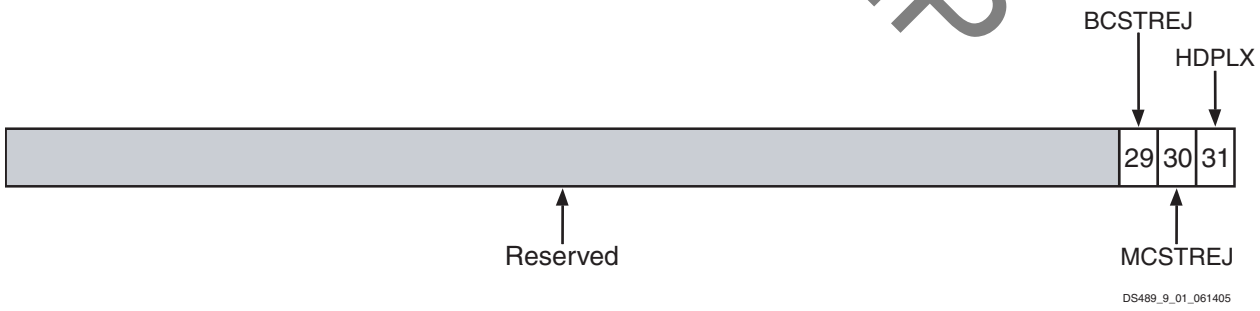

Figure 11: **PLB_TEMAC Control Register (offset 0x1000)**

Table 7: **PLB_TEMAC Control Register Bit Definitions**

Bit Location	Name	Core Access	Reset Value	Description
0 - 28	Reserved	Read	0x0	**Reserved**. These bits are reserved for future definition and will always return all zeros.
29	BCSTREJ	Read/Write	0x0	**BCSTREJ**. Indicates rejection of broadcast Ethernet frames. “0” - Accept broadcast frames. “1” - Reject broadcast frames.
30	MCSTREJ	Read/Write	0x0	**MCSTREJ**. Indicates rejection of multicast Ethernet frames. “0” - Accept multicast frames. “1” - Reject multicast frames.
31	HDPLX	Read/Write	0x0	**HDPLX**. Indicates half duplex operation for 10/100 mode. **(only full duplex is implemented in this release so this bit must always be ’0’)** “0” - Full duplex operation. “1” - Half duplex operation.

Transmit Packet Length Register (TPLR)

The transmit packet length register is actually a FIFO of register values each corresponding to a valid frame ready for transmit. The data for the frame is stored in the transmit data FIFO. The data is written to the PLB_TEMAC over the external processor bus interface either by simple DMA, Scatter/Gather DMA, or by direct memory mapped access. When presenting a transmit frame to the PLB_TEMAC, the frame data should first be written to the transmit data FIFO. The existence of data in the transmit packet length FIFO (FIFO empty flag is “0”) is used by the PLB_TEMAC to initiate the processing of transmit frames until this FIFO is empty.

This register can be read over the processor interface but only the PLB_TEMAC can remove a value from the FIFO. The PLB_TEMAC will remove the current length from the FIFO when it completes the corresponding transmission. If multiple reads are performed prior to that completion, the same value will be returned for each read operation. This register is wide enough to support 9K byte length jumbo frames (actually capable of up to 16K byte lengths).

Figure 12: **Transmit Packet Length Register (offset 0x1004)**

Table 8: **Transmit Packet Length Register Bit Definitions**

Bit Location	Name	Core Access	Reset Value	Description
0-17	Reserved	Read	0x0	**Reserved.** These bits are reserved for future definition and will always return all zeros.
18-31	TXPL	Read/Write	0x0	**Transmit Packet Length**. The number of bytes of the corresponding transmit frame stored in the transmit data FIFO.

Transmit Status Register (TSR)

The transmit status register is actually a FIFO of register values each corresponding to a frame transmission attempt. The bits in this register reflect the specific status of the corresponding transmit operation including the PLB_TEMAC settings which were applied to the transmit operation. Reading this register causes the current value to be removed from the FIFO. For this release only bit 31 is active.

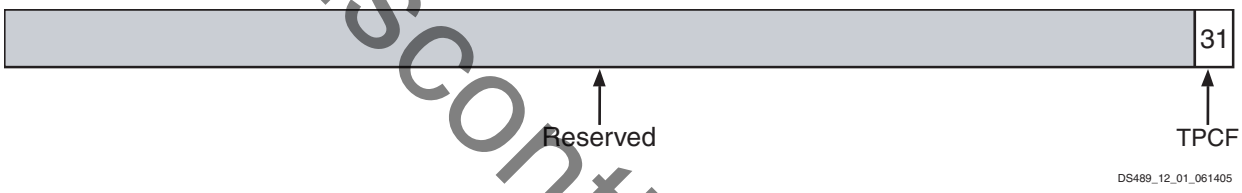

Figure 13: **Transmit Status Register (offset 0x1008)**

Table 9: **Transmit Status Register Bit Definitions**

Bit Location	Name	Core Access	Reset Value	Description
0 - 30	Reserved	Read	0x0	**Reserved**. These bits are reserved for future definition and will always return all zeros.
31	TPCF	Read	“1”	**Transmit Packet Complete Flag**. This bit is always “1” and is used to indicate to the software that a buffer descriptor associated with this frame has been completely processed by the DMA circuitry and is available for software use.

Receive Packet Length Register (RPLR)

The receive packet length register is actually a FIFO of register values each corresponding to a valid frame received. The data for the frame is stored in the receive data FIFO and the status word is stored in the receive status register FIFO.

The data is written by the PLB_TEMAC when the frame’s destination address passes the current address validation modes and when the frame has been determined to be valid and the receive data FIFO had enough locations that all of the frame data has been saved.

The existence of data in the receive packet length FIFO (FIFO empty flag is “0”) may be used to initiate the processing of received frames until this FIFO is empty. Reading this register causes the current value to be removed from the FIFO.

This register is wide enough to support 9K byte length jumbo frames (actually capable of up to 16K byte lengths).

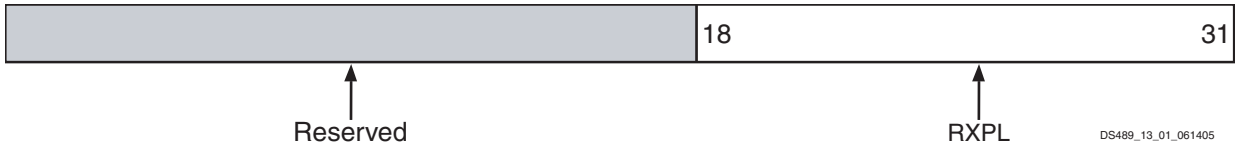

Figure 14: **Receive Packet Length Register (offset 0x100C)**

Table 10: **Receive Packet Length Register Bit Definitions**

Bit Location	Name	Core Access	Reset Value	Description
0-17	Reserved	Read	0x0	**Reserved.** These bits are reserved for future definition and will always return all zeros.
18-31	RXPL	Read	0x0	**Receive Packet Length**. The number of bytes of the corresponding receive packet stored in the receive data FIFO.

Receive Status Register (RSR)

The receive status register is a place holder for the receive status register that is used by the Scatter Gather DMA interface. The PLB_TEMAC does not need a receive status register but is required to provide the correct value in bit 31 to the generalized Scatter Gather DMA circuitry as part of a standard receive frame operation.

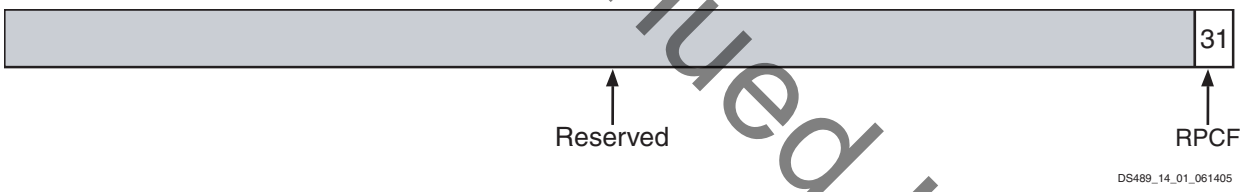

Figure 15: **Receive Status Register (offset 0x1010)**

Table 11: **Receive Status Register Bit Definitions**

Bit Location	Name	Core Access	Reset Value	Description
0 - 30	Reserved	Read	0x0	**Reserved**. These bits are reserved for future definition and will always return all zeros.
31	RPCF	Read	“0”	**Receive Packet Complete Flag**. This bit is always “1” and is used to indicate to the software that a buffer descriptor associated with this frame has been completely processed by the DMA circuitry and is available for software use.

Interframe Gap Register (IFG)

The Interframe Gap Register controls the duration of the interframe Gap (IFG). When enabled by the HARD_TEMAC transmit configuration register, ETXC bit 6 (ETXIFG), the value present in this register at the start of frame transmission determines the number of "idles" added to the minimum IFG between the end of the present frame and the start of transmission of the next frame.

An "idle" is defined as 8 Ethernet bus bit times. The minimum IFG is 96 bit times or 12 idles. A value of 0x1 in this register would add to the 12 idles making it 13 idles resulting in an IFG of 104 Ethernet bit times. In order to use the minimum allowed IFG which allows maximum data transfer on the Ethernet, a value of 0x0 should be used in this register.

This value is sampled each time the EMAC acknowledges the transmit client interface. When disabled by ETXIFG bit of TXC, the HARD_TEMAC will insert the IEEE 802.3-2002 default of 96 bit times.

Figure 16: **Interframe Gap Register (offset 0x1014)**

Table 12: **Interframe Gap Register Bit Definitions**

Bit Location	Name	Core Access	Reset Value	Description
0 - 23	Reserved	Read	0x0	**Reserved**. These bits are reserved for future definition and will always return all zeros.
24 - 31	IFGD	Read/Write	0x0	**Interframe Gap Data**. These bits denote the value of interframe gap delay. If this function is enabled by a configuration bit in the EMAC transmitter control register, then the value present in this register at the start of transmission will determine the number of clock cycles between subsequent frame transmissions. For example, if this register contains 0x0E, then there will be 14 transmit clock cycles between the end of the current frame and start of the next frame.

Transmit Pause Packet Register (TPPR)

The Transmit Pause Packet Register provides a value of pause when enabled by the Flow Control Configuration Register. When enabled, the EMAC will transmit a pause frame whenever this register is written. Pause values are defined in units of pause quanta which are defined as 512 bit times for the current transmission speed. Therefore, pause times may have values ranging from 0 to 65,536 * 512 bit times.

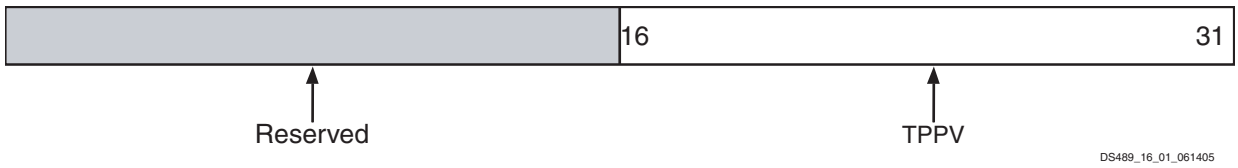

Figure 17: **Transmit Pause Packet Register (offset 0x1018)**

Table 13: **Transmit Pause Packet Register Bit Definitions**

Bit Location	Name	Core Access	Reset Value	Description
0 - 15	Reserved	Read	0x0	**Reserved**. These bits are reserved for future definition and will always return all zeros.
16 - 31	TPPV	Read/Write	0x0	**Transmit Pause Packet Value**. These bits denote the value of the transmit pause packet pause time in units of 512 bit times. If enabled by the EMAC flow control configuration register, writing a value into this register initiates the transmission of a single pause packet with the pause value defined in this bit field.

MII Management Control Register (MGTCR)

The MII Management Control Register is used with the MII management Data Register to perform read and writes between the PLB_TEMAC/HARD_TEMAC and the external PHY device via the MII management interface. **The MII Management interface is not implemented in this release.**

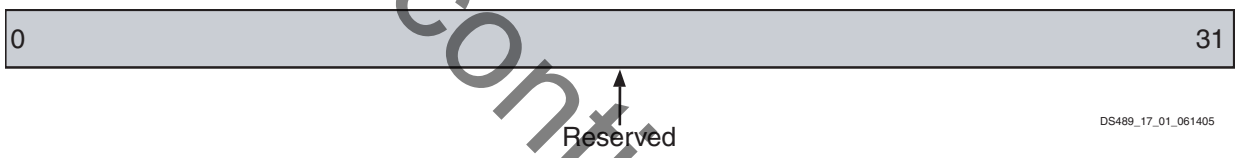

Figure 18: **MII Management Control Register (offset 0x101C)**

Table 14: **MII Management Control Register Bit Definitions**

Bit Location	Name	Core Access	Reset Value	Description
0 - 31	Reserved	Read	0x0	**Reserved**. These bits are reserved for future definition.

MII Management Data Register (MGTDR)

The MII Management Data Register is used with the MII management Control Register to perform read and writes between the PLB_TEMAC/HARD_TEMAC and the external PHY device via the MII management interface. **The MII Management interface is not implemented in this release.**

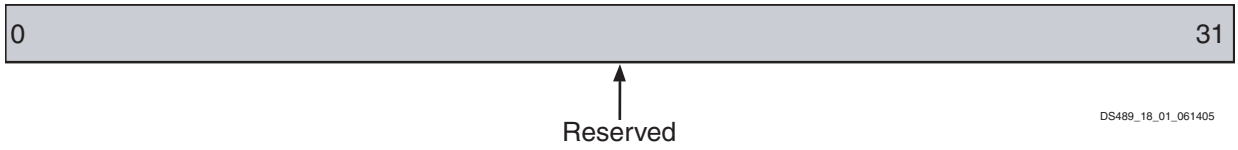

Figure 19: **MII Management Data Register (offset 0x1020)**

Table 15: **MII Management Data Register Bit Definitions**

Bit Location	Name	Core Access	Reset Value	Description
0 - 31	Reserved	Read	0x0	**Reserved**. These bits are reserved for future definition.

HARD_TEMAC Core Registers

Receiver Configuration Word 0 (RXC0)

Word 0 of the Receiver Configuration holds the 32 least significant bits of pause frame MAC address.

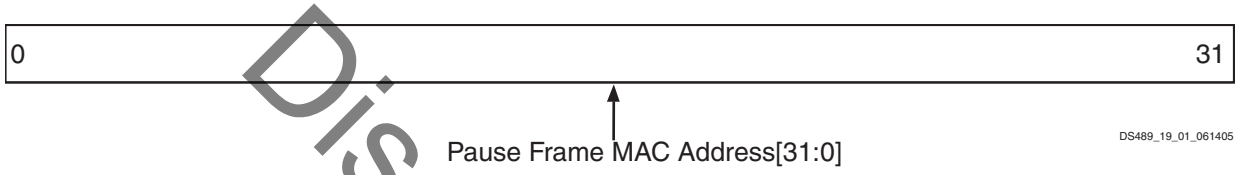

Figure 20: **Receiver Configuration Word 0 (offset 0x3200)**

Table 16: **Receiver Configuration Word 0 Register Bit Definitions**

Bit Location	Name	Core Access	Reset Value	Description
0 - 31	ERXC0	Read/Write	TIEEMACCONFIGVEC[31:0] 0x0	**Pause Frame MAC Address [31:0]**. This address is used by the HARD_TEMAC to match against the Destination Address of any incoming flow control frames. It is also used as the Source Address for any outbound flow control frames.

Receiver Configuration Word 1 (RXC1)

Word 1 of the Receiver Configuration holds the 16 most significant bits of pause frame MAC address and several enable and disable bits as defined below.

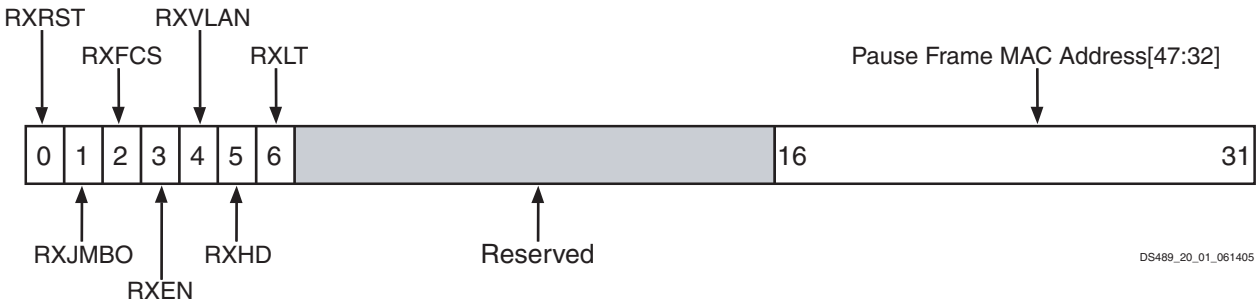

Figure 21: **Receiver Configuration Word 1 (offset 3240)**

Table 17: **Receiver Configuration Word 1 Register Bit Definitions**

Bit Location	Name	Core Access	Reset Value	Description
0	RXRST	Read/Write	TIEEMACCONFIGVEC[53] 0	**Receiver reset**. When the bit is set to “1”, the HARD_TEMAC transmitter will be reset. The bit will then automatically revert “0”. Note that this reset will also set all of the Receiver configuration registers to their default values.
1	RXJMBO	Read/Write	TIEEMACCONFIGVEC[52] 1	**Jumbo frame enable**. When this bit is set to “1”, the HARD_TEMAC receiver will accept frames larger than the IEEE802.3-2002 maximum legal length. When this bit is set to “0”, the core will only accept frames up to the specified maximum.
2	RXFCS	Read/Write	TIEEMACCONFIGVEC[51] 0	**In-band FCS enable**. When set to “1” the FCS field is passed to the client. Otherwise the FCS field is removed from the frame passed to the client. In both cases, the HARD_TEMAC will verify the frame FCS. **FCS removal is not implemented in this release so this bit should always be ’1’**.
3	RXEN	Read/Write	TIEEMACCONFIGVEC[50] 1	**Receiver enable**. If set to “1” the HARD_TEMAC receiver is enabled. Otherwise, the HARD_TEMAC ignores any activity on the RX port of the physical interface.
4	RXVLAN	Read/Write	TIEEMACCONFIGVEC[49] 1	**VLAN enable**. When set to “1” VLAN tagged frames will be accepted by the receiver.
5	RXHD	Read/Write	TIEEMACCONFIGVEC[48] 0	**Half Duplex**. If “1”, the receiver will operate in half duplex mode. **Half Duplex is not implemented in this release so this bit should always be ’0’.**
6	RXLT	Read/Write	TIEEMACCONFIGVEC[63] 0	**Length/type error check disable**. When this bit is set to “1”, the core will not perform the Length/type field error checks. When “0”. normal operation.
7 - 15	Reserved	Read	0x0	**Reserved**. These bits are reserved for future definition and will always return all zeros.
16 - 31	ERXC1	Read/Write	TIEEMACCONFIGVEC[47:32] 0x0	**Pause frame MAC Source Address [47:32]**. This address is used by the HARD_TEMAC to match against the Destination Address of any incoming flow control frames. It is also used as the Source Address for any outbound flow control frames.

Transmitter Configuration (TXC)

The Transmitter Configuration provides operational features in the transmit side of the HARD_TEMAC.

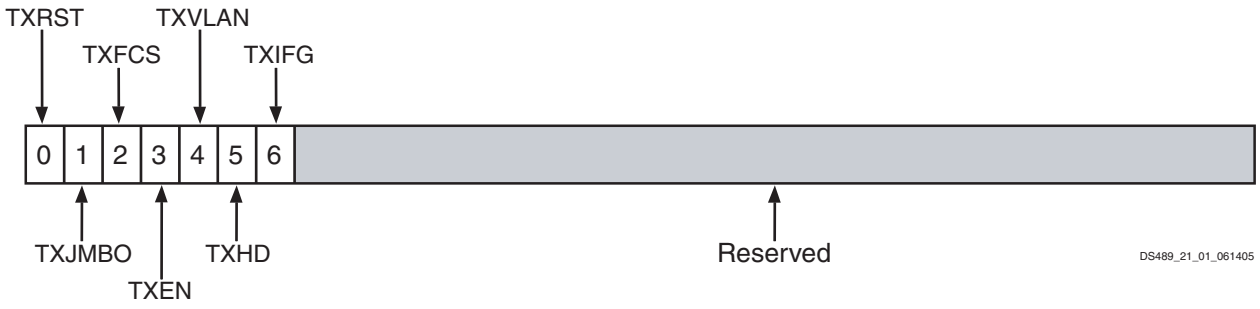

Figure 22: **Transmitter Configuration (offset 0x3280)**

Table 18: **Transmitter Configuration Register Bit Definitions**

Bit Location	Name	Core Access	Reset Value	Description
0	TXRST	Read/Write	TIEEMACCONFIGVEC[60] 0	**Transmitter reset**. When the bit is set to “1”, the HARD_TEMAC receiver will be reset. The bit will then automatically revert “0”. Note that this reset will also set all of the transmitter configuration register to default values.
1	TXJMBO	Read/Write	TIEEMACCONFIGVEC[59] 1	**Jumbo frame enable**. When this bit is set to “1”, the HARD_TEMAC transmitter will send frames larger than the IEEE802.3-2002 maximum legal length. When this bit is set to “0”, the core will only send frames up to the specified maximum.
2	TXFCS	Read/Write	TIEEMACCONFIGVEC[58] 0	**In-band FCS enable**. When set to “1” the transmitter will expect the FCS field to be passed from the client. When set “0”, the transmitter will append padding as required, and compute and append the FCS.
3	TXEN	Read/Write	TIEEMACCONFIGVEC[57] 1	**Transmitter enable**. If set to “1” the HARD_TEMAC transmitter is enabled. Otherwise, the core transmitter is disabled.
4	TXVLAN	Read/Write	TIEEMACCONFIGVEC[56] 1	**VLAN enable**. When set to “1” VLAN tagged frames will be sent by the transmitter.
5	TXHD	Read/Write	TIEEMACCONFIGVEC[55] 0	**Half Duplex**. If “1”, the transmitter will operate in half duplex mode. **Half Duplex is not implemented in this release so this bit should always be ’0’.**
6	TXIFG	Read/Write	TIEEMACCONFIGVEC[54] 0	**Interframe gap adjust enable**. When set to "1", the transmitter uses the value of the IFGP register at the start of frame transmission to adjust the Interframe Gap.
7 - 31	Reserved	Read	0x0	**Reserved**. These bits are reserved for future definition and will always return all zeros.

Flow Control Configuration (FCC)

The Flow Control Configuration register enables or disabled HARD_TEMAC flow control.

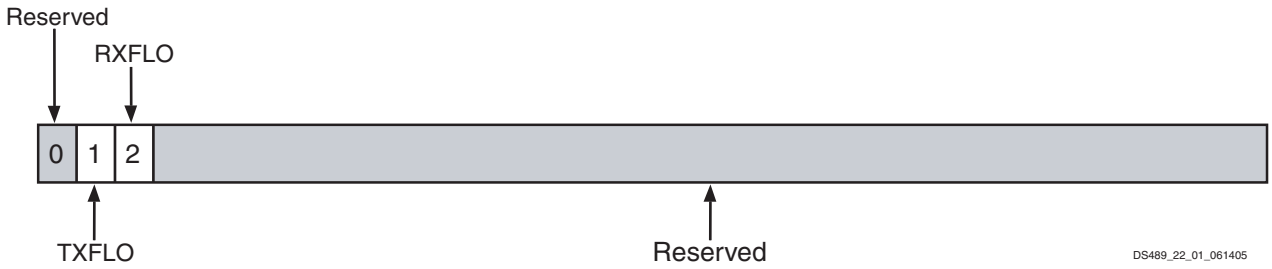

Figure 23: **Flow Control Configuration Register (offset 0x3232C0)**

Table 19: **Flow Control Configuration Register Bit Definitions**

Bit Location	Name	Core Access	Reset Value	Description
0	Reserved	Read/Write	0	**Reserved**. These bits are reserved for future definition and will always return all zeros.
1	TXFLO	Read/Write	TIEEMACCONFIGVEC[61] 1	**Transmit Flow Control Enable**. When “0”, the request to transmit pause packets (write to TPP register) is ignored. When “1”. requesting the transmit of a pause packet (write to TPP register) will cause the HARD_TEMAC to send a flow control frame.
2	RXFLO	Read/Write	TIEEMACCONFIGVEC[62] 1	**Receive Flow Control Enable**. When “0”, received flow control frames will be passed to the client. When “1”, received flow control frames will inhibit the HARD_TEMAC transmitter operation for a short period of time as defined in IEEE802.3-2002.
3 - 31	Reserved	Read	0x0	**Reserved**. These bits are reserved for future definition and will always return all zeros.

EMAC Mode Configuration Register (EMCFG)

The EMAC Mode Configuration register provides configuration status of link speeds and HARD_TEMAC PHY interface options as predefined at system build time based on PLB_TEMAC parameters (available in future releases that support PHY interfaces in addition to the GMII interface).

This release does not support the RGMII, SGMII, or 1000BaseX interfaces (it supports the GMII interface).

Bits 5 - 7 refer to an internal interface between the PLB_TEMAC and the HARD_TEMAC and are fixed and should not be needed by the user.

Bits 0 & 1 must be set to indicate the current operating link speed of the system. This may either be fixed or may use auto negotiation.

Since the MII management interface is not implemented in this release, another means must be provided for accessing auto negotiation results from the PHY device.

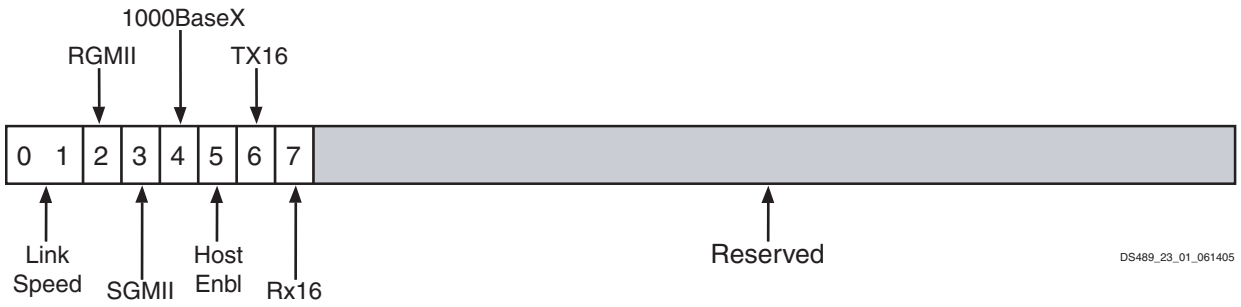

Figure 24: **EMAC Mode Configuration Register (offset 0x3300)**

Table 20: **EMAC Mode Configuration Register Bit Definitions**

Bit Location	Name	Core Access	Reset Value	Description
0 - 1	Link Speed	Read/Write	TIEEMACCONFIGVEC[72:71] 00	**Link Speed**. Determines link speed of operation: 00 = 10 Mb/s 01 = 100 Mb/s 10 = 1000 Mb/s 11 = N/A
2	RGMII	Read	TIEEMACCONFIGVEC[70] 0	**RGMII mode enable**. When set to “1”, RGMII is enabled.
3	SGMII	Read	TIEEMACCONFIGVEC[69] 0	**SGMII mode enable**. When set to “1”, SGMII is enabled.
4	1000BaseX	Read	TIEEMACCONFIGVEC[68] 0	**1000BaseX mode enable**. When this bit is set to “1”, the Ethernet MAC is configured in 1000Base-X mode.
5	Host Enable	Read	TIEEMACCONFIGVEC[67] 1	**Host Interface Enable**. When this bit is set to “1”, the host interface is used.
6	TX 16 Bit	Read	TIEEMACCONFIGVEC[66] 0	**Transmit 16-bit Client Interface enable**. When this bit is set to “1”, the transmit data client interface is 16 bits wide. When this bit is set to 0, the transmit data client interface is 8 bits wide.
7	RX 16 Bit	Read	TIEEMACCONFIGVEC[65] 0	**Receive 16-bit Client Interface enable**. When this bit is set to “1”, the receive data client interface is 16 bits wide. When this bit is set to 0, the receive data client interface is 8 bits wide.
8 - 31	Reserved	Read	0x0	**Reserved**. These bits are reserved for future definition and will always return all zeros.

RGMII/SGMII Configuration Register (GMIC)

The RGMII/SGMII Configuration register provides configuration status for HARD_TEMAC as shown.

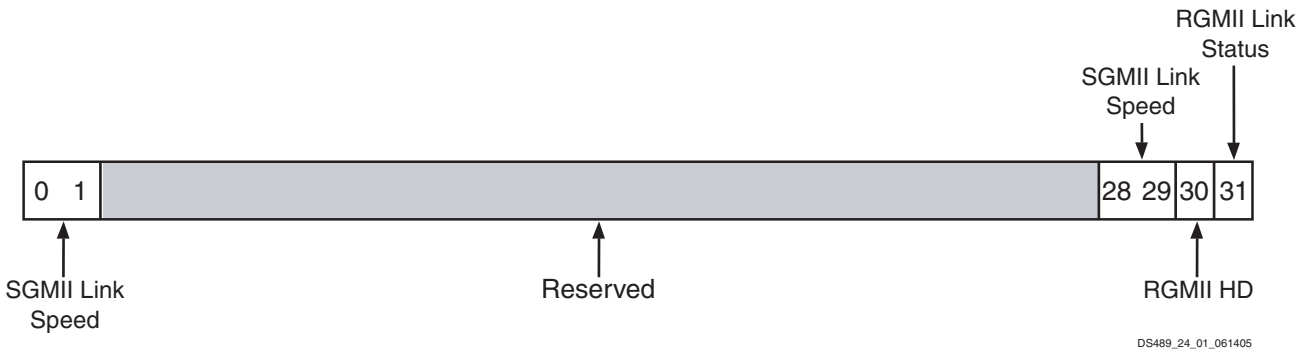

Figure 25: **RGMII/SGMII Configuration Register (offset 0x3320)**

Table 21: **RGMII/SGMII Configuration Register Bit Definitions**

Bit Location	Name	Core Access	Reset Value	Description
0 - 1	SGMII Link Speed	Read	00	**SGMII Link Speed**. Valid in SGMII mode configuration only. This displays the SGMII speed information, as received by TX_CONFIG_REG[11:10] in the PCS/PMA register. This 2-bit vector is defined with the following values: 00 = 10 Mb/s 01 = 100 Mb/s 10 = 1000 Mb/s 11 = N/A
2 -27	Reserved	Read	0x0	**Reserved**. These bits are reserved for future definition and will always return all zeros.

Table 21: **RGMII/SGMII Configuration Register Bit Definitions *(Continued)***

Bit Location	Name	Core Access	Reset Value	Description
28 - 29	RGMII Link Speed	Read	00	**RGMII Link Speed**. Valid in RGMII mode configuration only. Link information from PHY to HARD_TEMAC as encoded by GMII_RX_DV and GMII_RX_ER during the IFG. This 2-bit vector is defined with the following values: 00 = 10 Mb/s 01 = 100 Mb/s 10 = 1000 Mb/s 11 = N/A
30	RGMII Half Duplex	Read	0	**RGMII Half Duplex**. Valid in RGMII mode configuration only. When this bit is “1”, the HARD_TEMAC operates in half-duplex mode. When this bit is “0”, the core operates in full-duplex mode. This displays the duplex information from PHY to HARD_TEMAC, encoded by GMII_RX_DV and GMII_RX_ER during the IFG.
31	RGMII Link Status	Read	0	**RGMII Link Status**. Valid in RGMII mode configuration only. When this bit is “1”, the link is up. When this bit is “0”, the link is down. This displays the link information from PHY to HARD_TEMAC, encoded by GMII_RX_DV and GMII_RX_ER during the IFG.

Management Configuration Register (MC)

The Management Configuration Register provides control for the HARD_TEMAC PHY MII management (MDIO) interface. The MDIO interface supplies a clock to the external devices, EMAC#PHYMCLKOUT. This clock is derived from the HARD_TEMAC HOSTCLK signal which is connected to the PLB_CLK inside the PLB_TEMAC core using the value in the Clock Divide[5:0] configuration register. The frequency of the MDIO clock is given by the following equation:

$$f_{MDC} = \frac{f_{HOSTCLK}}{(1 + \text{Clock Divide}[5:0]) \times 2}$$

To comply with the IEEE 802.3-2002 specification for this interface, the frequency of EMAC#PHYMCLKOUT should not exceed 2.5 MHz. To prevent EMAC#PHYMCLKOUT from being out of specification, the Clock Divide[5:0] value powers up at 000000. While this value is in the register, it is impossible to enable the MDIO interface. Given this, even if the user has enabled the MDIO interface by setting bit 25 of this register, the MDIO port will still be disabled until a non-zero value has been written into the clock divide bits. **The MII management interface is not implemented in this release**.

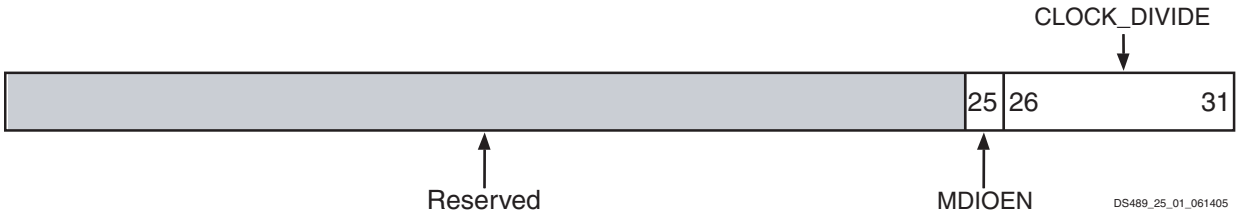

Figure 26: **Management Configuration Register (offset 0x3340)**

Table 22: **Management Configuration Register Bit Definitions**

Bit Location	Name	Core Access	Reset Value	Description
0 - 24	Reserved	Read	0x0	**Reserved**. These bits are reserved for future definition and will always return all zeros.
25	MDIO	Read / Write	TIEEMACCONFIGVEC[73] 0	**MII management enable**. When this bit is “1”, the MII management interface is used to access PHY devices. When this bit is 0, the MII management interface is disabled and the MDIO signal remain inactive.
26 - 31	CLK_DVD	Read / Write	0x0	**Clock Divide [5:0]**. This value is used to derive the EmacPhyMclkOut for external devices.

Unicast Address Register Word 0 (UAW0)

The Unicast Addresses Registers combine to provide a 48 bit ethernet station address. Word 0 provides the low order 32 bits of the address while word 1 provides the high order 16 bits.

Figure 27: **Unicast Address Register Word 0 (offset 0x3380)**

Table 23: **Unicast Address Register Word 0 Bit Definitions**

Bit Location	Name	Core Access	Reset Value	Description
0 - 31	UAW0	Read/Write	TIEEMACUNICASTADDR[31:0] 0x0	**MAC Unicast Address bits [31:0]**.

Unicast Address Register Word 1 (UAW1)

The Unicast Addresses Registers combine to provide a 48 bit ethernet station address. Word 0 provides the low order 32 bits of the address while word 1 provides the high order 16 bits.

Figure 28: **Unicast Address Register Word 1 (offset 0x3384)**

Table 24: **Unicast Address Register Word 1 Bit Definitions**

Bit Location	Name	Core Access	Reset Value	Description
0 - 15	Reserved	Read	0x0	**Reserved**. These bits are reserved for future definition and will always return all zeros.
16 - 31	UAW1	Read/Write	TIEEMACUNICASTADDR[47:32] 0x0	**MAC Unicast Address bits [47:32]**.

Multicast Address Register Word 0 (MAW0)

The Multicast Addresses Registers combine to provide a 48 bit ethernet addresses to store in content addressable memory (CAM). Word 0 provides the low order 32 bits of the address while word 1 provides the high order 16 bits. Word also provides CAM register addresses and the read or write control signal. **The multicast address filtering via the CAM is not implemented in this release.**

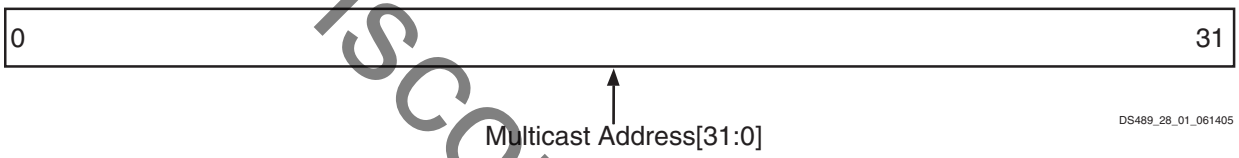

Figure 29: **Multicast Address Register Word 0 (offset 0x3388)**

Table 25: **Multicast Address Register Word 0 Bit Definitions**

Bit Location	Name	Core Access	Reset Value	Description
0 - 31	MAW0	Read/Write	0x0	**MAC Multicast Address bits [31:0]**.

Multicast Address Register Word 1 (MAW1)

The Multicast Addresses Registers combine to provide a 48 bit ethernet addresses to store in content addressable memory (CAM). Word 0 provides the low order 32 bits of the address while word 1 provides the high order 16 bits. Word also provides CAM register addresses and the read or write control signal. **The multicast address filtering via the CAM is not implemented in this release.**

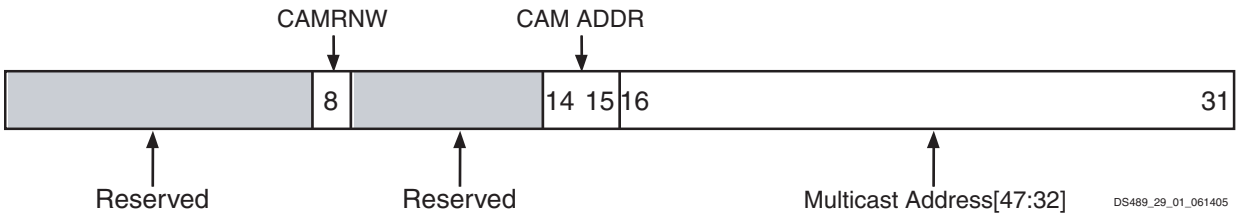

Figure 30: **Multicast Address Register Word 1 (offset 0x338C)**

Table 26: **Multicast Address Register Word 1 Bit Definitions**

Bit Location	Name	Core Access	Reset Value	Description
0 - 7	Reserved	Read/Write	0x0	**Reserved**. These bits are reserved for future definition and will always return all zeros.
8	CAMRNW	Read/Write	0	**CAMRNW**. CAM read, not write used to control the reading and writing of Multicast addresses into the content addressable memory registers.
9 - 13	Reserved	Read/Write	0x0	**Reserved**. These bits are reserved for future definition and will always return all zeros.
14 - 15	CAMADDR	Read/Write	0x0	**CAMADDR**. This two bit vector is used to choose the CAM Register to access. 00 = CAM Register 0 01 = CAM Register 1 10 = CAM Register 2 11 = CAM Register 3
16 - 31	MAW0	Read/Write	0x0	**MAC Multicast Address bits [47:32]**.

Address Filter Mode Register (AFM)

This is a one bit register used to enable or disable address filtering. When promiscuous mode is enabled, all inbound frames will be received and processed. When promiscuous mode is disabled, all inbound frames will be filtered by their respective destination addresses subject the present unicast, multicast, and broadcast addresses programmed into the MAC.

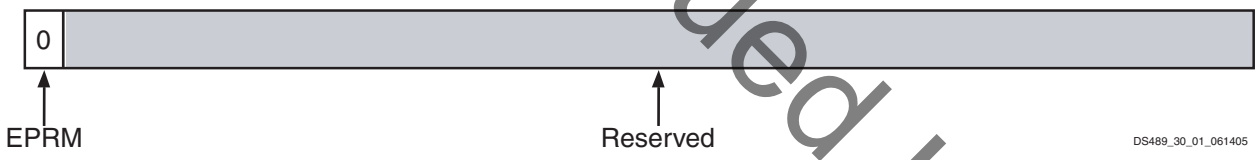

Figure 31: **Address Filter Mode Register (offset 0x3390)**

Table 27: **Address Filter Mode Register Bit Definitions**

Bit Location	Name	Core Access	Reset Value	Description
0	EPRM	Read / Write	TIEEMACCONFIGVEC[64] 1	**Promiscuous Mode Enable**. When this bit is set to “1”, the Address Filter Block is disable. When this bit is set to 0, the Address Filter Block is enabled.
1 - 31	Reserved	Read	0x0	**Reserved**. These bits are reserved for future definition and will always return all zeros.

Basic Usage

The PLB_TEMAC was designed to provide processor bus access to the HARD TEMACs to be used in embedded systems. Systems must be built through the Embedded Development Kit to attach the PLB_TEMAC, the HARD_TEMAC, multi gigabit transceivers, processor, memory, busses, clocking and addition embedded components.

The section is not intended to describe the building of systems, EDK documents describe this process. This section briefly describes very simple operation of the PLB_TEMAC through register accesses.

To initialize the PLB_TEMAC system receive and transmit, RXC1 and TXC registers may both be write with hex value 0x80000000. This will reset both HARD_TEMAC receive and transmit paths. Flow control may also be reset by writing the same hex value 0x80000000 to FCC register. After reset, the receive and transmit paths should be configured by writing to RXC1 and TXC. Flow control should also be configured at this time, if it is desired.

The unicast address is configured by writing to UAW1 and UAW0. For a unicast address of 0x060504030201, write UAW1= 0x00000102 and UAW0 = 0x03040506 due to the reverse byte order transmission of Ethernet. CFG then needs to be written to configure the PLB_TEMAC system as appropriate. Finally, promiscuous reception may be disabled by writing AFM = 0x00000000.

After configuring the PLB_TEMAC system, frames will automatically be transmitted by writing frame data to the Write Packet FIFO followed by writing a length in the TPLR. This provides the very simple transmission through a polling mode. After the frame transmits, the transmit status may be obtained from TSR.

Receiving Ethernet frames works in a fashion similar to transmission, although the steps are reversed. Polling of the RSR, RPLR or Read Packet FIFO status indicates reception of an Ethernet frame. Reading the RPLR provides the frame length in bytes and the RSR provides received frame status. Given the number of received frame bytes, the appropriate number of word of double word reads of the Read Packet FIFO will provide the frame data.

Figure 28 illustrates a very simple initialization, transmission and reception of a single Ethernet frame. Refer to the register definitions for further information and options. Additional register writes and reads are required to perform interrupts and DMA operations.

Table 28: **PLB_TEMAC Simple Usage**

Register	Access	Value	Activity
RXC1	Write Word	0x80000000	Reset EMAC Rx
TXC	Write Word	0x80000000	Reset EMAC Tx
FCC	Write Word	0x80000000	Reset EMAC Flow Control
RXC1	Write Word	0x10000000	Enable EMAC Rx, no VLAN, no Jumbo, Strip FCS
TXC	Write Word	0x10000000	Enable EMAC Tx, no VLAN, no Jumbo, no client supplied FCS
UAW1	Write Word	0x00000102	EMAC Unicast Address 0x060504030201
UAW0	Write Word	0x03040506	EMAC Unicast Address 0x060504030201
CFG	Write Word	0x84000000	EMAC Enables 1G Speed and Host Interface
AFM	Write Word	0x00000000	EMAC Enables Address Filtering
WPFIFO_DATA	Write Double	0x0605040302010605	6 bytes of destination address, 2 bytes source address
WPFIFO_DATA	Write Double	0x040302010042FEFF	4 bytes source address, 2 bytes length, 2 bytes frame data
WPFIFO_DATA	Write Double	0x0001020304050607	8 bytes of frame data
WPFIFO_DATA	Write Double	0x08090A0B0C0D0E0F	8 bytes of frame data
WPFIFO_DATA	Write Double	0x1011121314151617	8 bytes of frame data
WPFIFO_DATA	Write Double	0x18191A1B1C1D1E1F	8 bytes of frame data
WPFIFO_DATA	Write Double	0x2021222324252627	8 bytes of frame data
WPFIFO_DATA	Write Double	0x28292A2B2C2D2E2F	8 bytes of frame data
WPFIFO_DATA	Write Double	0x3031323334353637	8 bytes of frame data
WPFIFO_DATA	Write Double	0x38393A3B3C3D3E3F	8 bytes of frame data
TPLR	Write Word	0x00000050	Transmit length (0x50 = 80 bytes), this starts transmission

Table 28: **PLB_TEMAC Simple Usage *(Continued)***

TSR	Read Word	0x00000001	A typical value after Tx Complete is indicated by interrupt
RSR	Read Word	0x00000001	A typical value after Rx Complete is indicated by interrupt
RPLR	Read Word	0x00000050	Receive length (0x50 = 80 bytes) indicates no. bytes to read
RPFIFO_DATA	Read Double	0x0605040302010605	6 bytes of destination address, 2 bytes source address
RPFIFO_DATA	Read Double	0x040302010042FEFF	4 bytes source address, 2 bytes length, 2 bytes frame data
RPFIFO_DATA	Read Double	0x0001020304050607	8 bytes of frame data
RPFIFO_DATA	Read Double	0x08090A0B0C0D0E0F	8 bytes of frame data
RPFIFO_DATA	Read Double	0x1011121314151617	8 bytes of frame data
RPFIFO_DATA	Read Double	0x18191A1B1C1D1E1F	8 bytes of frame data
RPFIFO_DATA	Read Double	0x2021222324252627	8 bytes of frame data
RPFIFO_DATA	Read Double	0x28292A2B2C2D2E2F	8 bytes of frame data
RPFIFO_DATA	Read Double	0x3031323334353637	8 bytes of frame data
RPFIFO_DATA	Read Double	0x38393A3B3C3D3E3F	8 bytes of frame data

Design Implementation

Design Tools

The PLB_TEMAC design is implemented using VHDL code.

Xilinx XST is the synthesis tool used for synthesizing the PLB_TEMAC. The NGC netlist output is then input to the Xilinx Foundation tool suite for device implementation.

Target Technology

The intended target technologies is Virtex4-FX FPGAs.

Device Utilization and Performance Benchmarks

Since the PLB_TEMAC is a module that will be used with other design pieces in the FPGA, the utilization and timing numbers reported in this section are just estimates. As the PLB_TEMAC is combined with other pieces of the FPGA design, the

utilization of FPGA resources and timing of the PLB_TEMAC design will vary from the results reported here. The PLB_TEMAC benchmarks are shown in Table 29 for a Virtex4-FX FPGA.

Table 29: **PLB_TEMAC FPGA Performance and Resource Utilization Benchmarks (Virtex4-FX)**

Parameter Values							Device Resources				fMAX (MHz)
C_IPIF_RDFIFO_DEPTH C_IPIF_WRFIFO_DEPTH	C_MAC_FIFO_DEPTH	C_INCLUDE_DRE	C_INCLUDE_DEV_MIR	C_INCLUDE_RESET	C_INCLUDE_DEV_PENCODER	C_DMA_TYPE	Slices	Slice Flip-Flops	BRAMS	4-input LUTs	
16384	16	0	0	0	0	1	1111	1145	4	1492	
32768	16	0	0	0	0	1	1120	1157	4	1508	
32768	16	0	0	0	1	1	1128	1157	4	1521	
32768	16	0	0	1	1	1	1141	1166	4	1534	
32768	16	0	1	1	1	1	1149	1168	4	1547	
32768	16	1	1	1	1	1	1423	1392	4	1966	
16384	16	0	0	0	0	3	2472	2123	4	3849	
32768	16	1	1	1	1	2	2572	2288	4	3948	
16384	16	1	0	0	0	3	2745	2347	4	4255	
16384	32	1	0	0	0	3	2773	2351	4	4303	
32768	16	1	1	1	1	3	2786	2370	4	4317	
262144	64	1	1	1	1	3	2859	2414	32	4481	

Notes:
1. ISE H.41

Specification Exceptions

The PLB_TEMAC design currently has no exceptions to the mandatory IEEE Std. 802.3 MII interface requirements.

Reference Documents

The following document contains reference information important to understanding the PLB_TEMAC design:

- IEEE Std. 802.3-2000
- Virtex-4 Embedded Tri-Mode Ethernet MAC User Guide (**UG074**)
- EDK Processor IP Reference Guide

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
6/30/05	1.0	Initial Xilinx release.