HARD Tri-Mode Ethernet MAC (TEMAC) (v3.00b)

DS501 April 24, 2009 **Product Specification**

Introduction

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

The HARD_TEMAC described in this document has been designed incorporating the applicable features described in *IEEE Std. 802.3-2002*. Differences between that specification and the Xilinx HARD_TEMAC implementation are highlighted and explained in the <RD Red>Specification Exceptions section.

The HARD_TEMAC is an intellectual property (IP) soft core designed for implementation in a Virtex®-4 FX FPGA. The HARD_TEMAC soft core provides a wrapper around the Hard TEMAC component implemented in the Virtex-4 FX FPGA silicon to allow it to be used in an embedded system with the Embedded Development Kit (EDK) tools. The Virtex-4 FX FPGA Hard TEMAC silicon component has a detailed users guide which should be used to supplement this document. See the "Reference Documents" section of this document.

The HARD_TEMAC v3.00b core is designed to be used with the PLB_TEMAC v3.00a core to couple the TEMAC to the PowerPC® controller via the Processor Local Bus (PLB).

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

LogiCORE™ Facts		
Core Specifics		
Supported Device Family	See EDK Supported Device Families.	
Version of Core	HARD_TEMAC	v3.00b
Resources Used		
	Min	Max
Total Core I/O	332	332
Core FPGA IOBs	4	48
LUTs	0	0
FFs	0	0
Block RAMs	0	0
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	See Tools for requirements.	
Verification		
Simulation		
Synthesis		
Support		
Provided by Xilinx, Inc.		

Features

- Filtering of “bad” receive frames to reduce processor bus utilization.
- Hardware selectable DCR or PLB host interface to configuration registers.
- GMII and MII interfaces to external PHY devices.
- SGMII supported through MGT interface to external copper PHY layer.
- Complies with IEEE 802.3-2000 specification.
- Full duplex operation.
- Media Independent Interface Management (MIIM) for access to PHY transceiver registers.
- 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.
- 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.
- Optional support of jumbo frames.

Functional Description

The HARD Tri-Mode Ethernet MAC (TEMAC) core is described in the sections and figures detailed below.

PLB Tri-mode EMAC System Overview

A PLB Tri-mode Ethernet System includes either one or two PLB_TEMAC soft cores for each HARD_TEMAC soft core that will be used.

HARD_TEMAC Silicon Component

The Hard TEMAC is a silicon component of each Virtex-4 FX FPGA. The HARD_TEMAC soft core (wrapper) enables the use of the Hard TEMAC silicon component in EDK embedded systems. Each Hard TEMAC silicon component 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(1), 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. Note that some features supported by the Hard TEMAC silicon component are not supported by the soft PLB_TEMAC/HARD_TEMAC cores implementation. Please refer to the "Features" list of this document.

1. IEEE Std. 802.3 uses the terms Frame and Packet interchangeably when referring to the Ethernet unit of transmission.

The Hard TEMAC silicon component functionality consumes no FPGA programmable resources since the Hard TEMAC is built into the silicon of each Virtex-4 FX FPGA. Please refer to the HARD_TEMAC silicon component specification for more details.

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

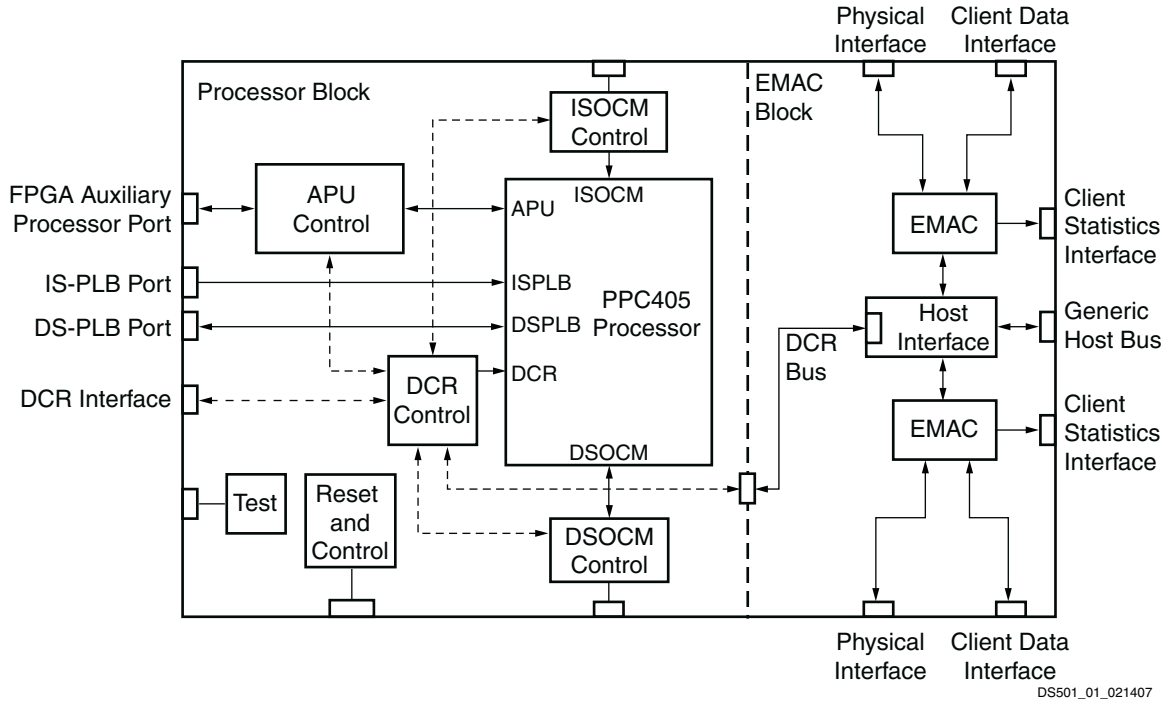

Figure 1: **Hard TEMAC and PowerPC Processor Controller Silicon Components Block Diagram**

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 controller and the Hard TEMAC. Each EMAC has its own set of Client, PHY, PHY Management, and statistics interfaces.

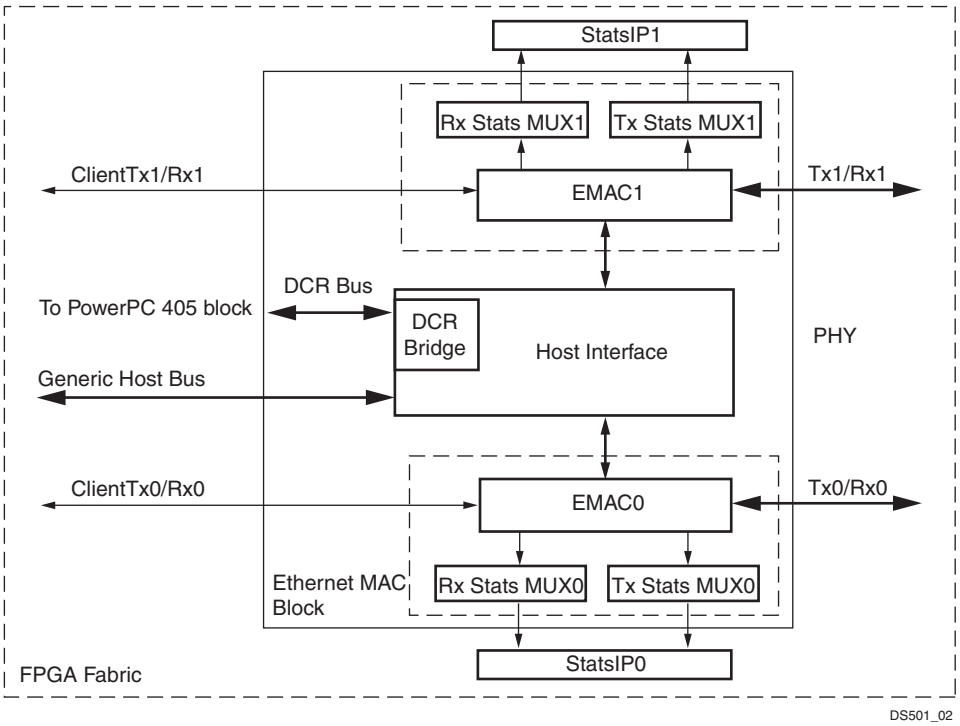

Figure 2: **Detailed Block Diagram of the Hard TEMAC Silicon Component**

PLB_TEMAC Core

The PLB_TEMAC provides access to the HARD_TEMAC host interface from the PLB. The DCR Interface, built in the silicon, is not supported.

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 for a PLB_TEMAC with DMA.

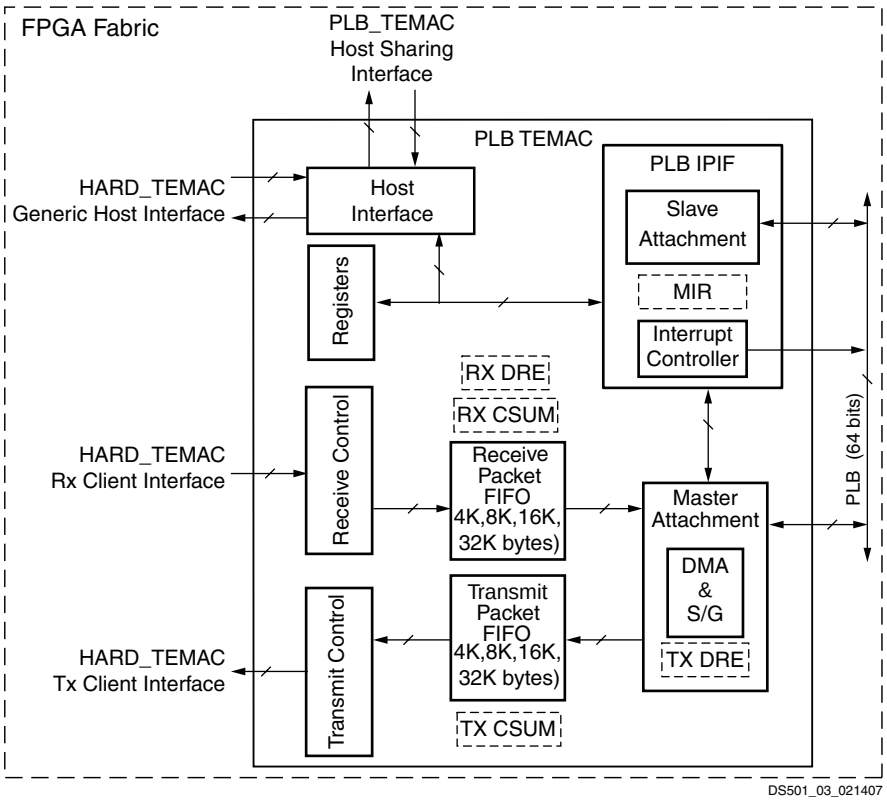

Figure 3: **PLB TEMAC Block Diagram**

HARD_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.

HARD_TEMAC Design Parameters

To allow the user to generate a HARD_TEMAC that is tailored for their system, certain features are parameterizable in the HARD_TEMAC design. This allows the user to have a design that only utilizes the resources required by their system and runs at the best possible performance. The features that are parameterizable in the Xilinx HARD_TEMAC design are shown in Figure 1

Table 1: **HARD_TEMAC Design Parameters**

Generic	Feature / Description	Parameter Name	Allowable Values	Default Value	VHDL Type
Top Level					
G1	PHY Interface Type	C_PHY_TYPE	0 = MII 1 = GMII 2 = RGMII v1.3 4 = SGMII	1	integer
G2	Enable Use of EMAC 1	C_EMAC1_ PRESENT	1 = EMAC1 Used 0 = EMAC1 Not Used	0	integer
G3	PHY Address for EMAC 0	C_TEMAC0_ PHYADDR	00001 - 11111 (1)	00001	std_logic_ vector
G4	PHY Address for EMAC 1	C_TEMAC1_ PHYADDR	00001 - 11111	00010	std_logic_ vector
Other					
G5	Provides a means for BSB to pass LOC coordinates for IDELAYCTRLs for a given board to EDK and is optional for user to set LOC constraints. This parameter is only used for RGMII PHY type.	C_IDELAYCT RL_LOC	<RD Red>See "Detailed Parameter Descriptions" on page 6.	NOT_SET	string
G6	Provides a means for BSB to pass a delay value for a given board to EDK and is optional for user to set delay constraints. This parameter is only used for RGMII PHY type.	C_RGMII_RX _CLK_DELAY	<RD Red>See "Detailed Parameter Descriptions" on page 6.	0	string

Notes:
1. The value "00000" is a broadcast PHY address and should not be used to avoid contention between the internal HARD_TEMAC PHY and the external PHY(s).The value "00000" is a broadcast PHY address and should not be used to avoid contention between the internal HARD_TEMAC PHY and the external PHY(s).
2.

Allowable Parameter Combinations

Each HARD_TEMAC includes two EMACs. If only one EMAC is used it must be EMAC 0 and C_EMAC1_PRESENT should be set to ’0’.

Detailed Parameter Descriptions

C_PHY_TYPE

The PHY Type parameter selects which PHY interface will be used by EMAC 0 (and EMAC 1 if present).

C_EMAC1_PRESENT

This parameter is used to indicate if both EMACs in the HARD_TEMAC are used or not used. If this parameter is set to ’1’ then both EMACs will be used otherwise only EMAC 0 will be used.

C_TEMAC0_PHYADDR and C_TEMAC1_PHYADDR

These values are used to control MII Management accesses to the internal PHY registers of EMAC 0 and EMAC 1 of the HARD_TEMAC core. An address of "00000" should not be used since this is a broadcast addresses and all PHYs will respond to MII Management requests to this address possibly causing contention. In most systems at least one and possibly more external PHYs will also be used. All PHYs should all have unique PHY addresses.

C_IDELAYCTRL_LOC

When a PHY type of RGMII is selected, an IDELAY primitive is used to help align the receive data with the receive clock. If both EMACs are used, one IDELAY primitive is used for each of the two EMACs. When IDELAY primitives are used, a IDELAYCTRL primitive is also required. The IDELAYCTRL primitive(s) must be located in the proper area in the silicon in order for it to be effective and this is accomplished by adding constraints to the ucf-file. Additionally, a 200 MHz clock must be supplied to the input RefClk which is used by the IDELAY and IDELAYCTRL primitives.

When two IDELAYCTRL primitives are used (when both EMACs are used in RGMII mode), LOC constraints are required on each primitive. The FPGA Editor tool can be helpful to determine IDELAYCTRL LOC coordinates for the user's pinout.

The method for setting the LOC constraint(s) is to use the C_IDELAYCTRL_LOC parameter. This parameter when properly set will generate constraints in the hard_temac core ucf file. Note that if the LOC constraints are set in the system top-level ucf-file, then this parameter has no effect since the constraints in the system top-level ucf-file override those in lower level ucf files.

The syntax of the parameter value is IDELAYCTRL_XNYM where N and M are coordinates and multiple entries are concatenated by - (i.e, dash). The first value corresponds to EMAC0 and the second value if present corresponds to EMAC1. The following is an example of how the parameter might be set in the MHS file when both EMACs are used in RGMII mode. The X and Y values will be different for each implementation. Please refer to the Virtex-4 User Guide for more information on selecting the correct IDELAY Controller location.

```
PARAMETER C_IDELAYCTRL_LOC="IDELAYCTRL_X0Y0-IDELAYCTRL_X0Y1"
```

The quotes are optional.

C_RGMII_RX_CLK_IDELAY

When a PHY type of RGMII is selected, an IDELAY primitive is used to align the receive data with the receive clock. The delay value must be provided for each IDELAY primitive used (one for each EMAC used).

The method for setting the delay value is to use the C_RGMII_RX_CLK_DELAY parameter. This parameter when properly set will generate constraints in the hard_temac core ucf file. Note that if the delay constraints are set in the system top-level ucf file, then this parameter has no effect since the constraints in the system top-level ucf file override those in lower level ucf files.

The syntax of the parameter value is N where N is the delay value for the receive clock and multiple entries are concatenated by - (i.e, dash). The first value corresponds to EMAC0 and the second value if present corresponds to EMAC1. The following is an example of how the parameter might be set in the MHS file when both EMACs are used in RGMII mode.

```
PARAMETER C_RGMII_RX_DELAY="20-20"
```

The quotes are optional.

HARD_TEMAC I/O Signals

The HARD_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 is shown in the PLB_TEMAC specification. In the signal list below, those signals that are assigned to the virtual bus are designated with an interface value of V4EMACDST0 and V4EMACDST1.

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

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

Port	Signal Name	Interface	I/O	Description
EMAC 0 Client Receive Interface				
P1	Emac0ClientRxd(7:0)	V4EMACDST0	O	Client receive Data
P2	Emac0ClientRxdVld	V4EMACDST0	O	Client receive data valid
P3	Emac0ClientRxdVldMsw	V4EMACDST0	O	Client receive data valid on most significant word
P4	Emac0ClientRxGoodFrame	V4EMACDST0	O	Client valid receive frame indicator
P5	Emac0ClientRxBadFrame	V4EMACDST0	O	Client invalid receive frame indicator
P6	Emac0ClientRxFrameDrop	V4EMACDST0	O	Client receive frame dropped indication
P7	Emac0ClientRxdVreg6	V4EMACDST0	O	Client receive data valid early registration
P8	Emac0ClientRxStats(6:0)	V4EMACDST0	O	Client receive statistics
P9	Emac0ClientRxStatsVld	V4EMACDST0	O	Client receive statistics valid indicator
P10	Emac0ClientRxStatsByteVld	V4EMACDST0	O	Client receive statistics byte valid
EMAC 0 Client Transmit Interface				

Table 2: **HARD_TEMAC I/O Signals *(Cont'd)***

Port	Signal Name	Interface	I/O	Description
P11	ClientEmac0Txd(7:0)	V4EMACDST0	I	Client transmit data
P12	ClientEmac0TxdVld	V4EMACDST0	I	Client transmit data valid
P13	ClientEmac0TxdVldMsw	V4EMACDST0	I	Client transmit data valid most significant word
P14	Emac0ClientTxAck	V4EMACDST0	O	Client transmit acknowlege
P15	ClientEmac0TxUnderRun	V4EMACDST0	I	Client tx under run
P16	Emac0ClientTxCollision	V4EMACDST0	O	Client transmit collision indicator
P17	Emac0ClientTxRetransmit	V4EMACDST0	O	Client retransmit indication
P18	ClientEmac0TxIfgDelay(7:0)	V4EMACDST0	I	Client interframe gap delay for tx.
P19	ClientEmac0TxFirstByte	V4EMACDST0	I	Client transmit first byte indicator
P20	Emac0ClientTxStats	V4EMACDST0	O	Client transmit statistics
P21	Emac0ClientTxStatsVld	V4EMACDST0	O	Client transmit statistics valid
P22	Emac0ClientTxStatsByteVld	V4EMACDST0	O	Client transmit statistics byte valid
EMAC 0 Control Interface				
P23	ClientEmac0PauseReq	V4EMACDST0	I	Pause request
P24	ClientEmac0PauseVal(15:0)	V4EMACDST0	I	Pause value
Emac 0 Clocks				
P25	GTX_Clk_0	CLK0	I	
P26	Rx_Client_Clk_0	V4EMACDST0	O	Receive client clock
P27	Tx_Client_Clk_0	V4EMACDST0	O	Transmit client clock
EMAC 0 MII Interface				
P28	MII_TxD_0(3:0)	PHY0	O	MII transmit data
P29	MII_Tx_En_0	PHY0	O	MII transmit enable
P30	MII_Tx_Er_0	PHY0	O	MII transmit error
P31	MII_RxD_0(3:0)	PHY0	I	MII receive data
P32	MII_Rx_Dv_0	PHY0	I	MII receive data valid
P33	MII_Rx_Er_0	PHY0	I	MII receive error
P34	MII_Rx_Clk_0	PHY0	I	MII receive clock
EMAC 0 MII & GMII Interface				
P35	MII_Tx_Clk_0	PHY0	I	MII and GMII transmit clock
EMAC 0 MII & GMII Interface				
P36	GMII_TxD_0(7:0)	PHY0	O	GMII transmit data
P37	GMII_Tx_En_0	PHY0	O	GMII transmit enable
P38	GMII_Tx_Er_0	PHY0	O	GMII transmit error

Table 2: **HARD_TEMAC I/O Signals *(Cont'd)***

Port	Signal Name	Interface	I/O	Description
P39	GMII_Tx_Clk_0	PHY0	O	GMII transmit clock
P40	GMII_RxD_0(7:0)	PHY0	I	GMII receive data
P41	GMII_Rx_Dv_0	PHY0	I	GMII receive data valid
P42	GMII_Rx_Er_0	PHY0	I	GMII receive error
P43	GMII_Rx_Clk_0	PHY0	I	GMII receive clock
EMAC 0 SGMII Interface				
P44	TxP_0	PHY0	O	SGMII transmit positive
P45	TxN_0	PHY0	O	SGMII transmit negative
P46	RxP_0	PHY0	I	SGMII receive positive
P47	RxN_0	PHY0	I	SGMII receive negative
EMAC 0 RGMII Interface				
P48	RGMII_TxD_0(3:0)	PHY0	O	Reserved for future use
P49	RGMII_Tx_Ctl_0	PHY0	O	Reserved for future use
P50	RGMII_TxC_0	PHY0	O	Reserved for future use
P51	RGMII_RxD_0(3:0)	PHY0	I	Reserved for future use
P52	RGMII_Rx_Ctl_0	PHY0	I	Reserved for future use
P53	RGMII_RxC_0	PHY0	I	Reserved for future use
EMAC 0 MDIO Interface				
P54	MdC_0	PHY0	O	MII Management Clock
P55	MdIO_0	PHY0	I/O	MII Management Data
P56	EMAC0ClientAnInterrupt	V4EMACDST0	O	Auto negotiation complete interrupt
EMAC 1 Client Receive Interface				
P57	Emac1ClientRxd(7:0)	V4EMACDST1	O	Client receive Data
P58	Emac1ClientRxdVld	V4EMACDST1	O	Client receive data valid
P59	Emac1ClientRxdVldMsw	V4EMACDST1	O	Client receive data valid on most significant word
P60	Emac1ClientRxGoodFrame	V4EMACDST1	O	Client valid receive frame indicator
P61	Emac1ClientRxBadFrame	V4EMACDST1	O	Client invalid receive frame indicator
P62	Emac1ClientRxFrameDrop	V4EMACDST1	O	Client receive frame dropped indication
P63	Emac1ClientRxdVreg6	V4EMACDST1	O	Client receive data valid early registration
P64	Emac1ClientRxStats(6:0)	V4EMACDST1	O	Client receive statistics
P65	Emac1ClientRxStatsVld	V4EMACDST1	O	Client receive statistics valid indicator
P66	Emac1ClientRxStatsByteVld	V4EMACDST1	O	Client receive statistics byte valid
EMAC 1 Client Transmit Interface				

Table 2: **HARD_TEMAC I/O Signals *(Cont'd)***

Port	Signal Name	Interface	I/O	Description
P67	ClientEmac1Txd(7:0)	V4EMACDST1	I	Client transmit data
P68	ClientEmac1TxdVld	V4EMACDST1	I	Client transmit data valid
P69	ClientEmac1TxdVldMsw	V4EMACDST1	I	Client transmit data valid most significant word
P70	Emac1ClientTxAck	V4EMACDST1	O	Client transmit acknowlege
P71	ClientEmac1TxUnderRun	V4EMACDST1	I	Client tx under run
P72	Emac1ClientTxCollision	V4EMACDST1	O	Client transmit collision indicator
P73	Emac1ClientTxRetransmit	V4EMACDST1	O	Client retransmit indication
P74	ClientEmac1TxIfgDelay(7:0)	V4EMACDST1	I	Client interframe gap delay for tx.
P75	ClientEmac1TxFirstByte	V4EMACDST1	I	Client transmit first byte indicator
P76	Emac1ClientTxStats	V4EMACDST1	O	Client transmit statistics
P77	Emac1ClientTxStatsVld	V4EMACDST1	O	Client transmit statistics valid
P78	Emac1ClientTxStatsByteVld	V4EMACDST1	O	Client transmit statistics byte valid
EMAC 1 Control Interface				
P79	ClientEmac1PauseReq	V4EMACDST1	I	Pause request
P80	ClientEmac1PauseVal(15:0)	V4EMACDST1	I	Pause value
Emac 1 Clocks				
P81	GTX_Clk_1	CLK1	I	
P82	Rx_Client_Clk_0	V4EMACDST1	O	Receive client clock
P83	Tx_Client_Clk_0	V4EMACDST1	O	Transmit client clock
EMAC 1 MII Interface				
P84	MII_TxD_1(3:0)	PHY1	O	MII transmit data
P85	MII_Tx_En_1	PHY1	O	MII transmit enable
P86	MII_Tx_Er_1	PHY1	O	MII transmit error
P87	MII_RxD_1(3:0)	PHY1	I	MII receive data
P88	MII_Rx_Dv_1	PHY1	I	MII receive data valid
P89	MII_Rx_Er_1	PHY1	I	MII receive error
P90	MII_Rx_Clk_1	PHY1	I	MII receive clock
EMAC 1 MII & GMII Interface				
P91	MII_Tx_Clk_1	PHY1	I	MII and GMII transmit clock
EMAC 1 GMII Interface				
P92	GMII_TxD_1(7:0)	PHY1	O	GMII transmit data
P93	GMII_Tx_En_1	PHY1	O	GMII transmit enable
P94	GMII_Tx_Er_1	PHY1	O	GMII transmit error

Table 2: **HARD_TEMAC I/O Signals *(Cont'd)***

Port	Signal Name	Interface	I/O	Description
P95	GMII_Tx_Clk_1	PHY1	O	GMII transmit clock
P96	GMII_RxD_1(7:0)	PHY1	I	GMII receive data
P97	GMII_Rx_Dv_1	PHY1	I	GMII receive data valid
P98	GMII_Rx_Er_1	PHY1	I	GMII receive error
P99	GMII_Rx_Clk_1	PHY1	I	GMII receive clock
EMAC 1 SGMII Interface				
P100	TxP_1	PHY1	O	SGMII transmit positive
P101	TxN_1	PHY1	O	SGMII transmit negative
P102	RxP_1	PHY1	I	SGMII receive positive
P103	RxN_1	PHY1	I	SGMII receive negative
EMAC 1 RGMII Interface				
P104	RGMII_TxD_1(3:0)	PHY1	O	Reserved for future use
P105	RGMII_Tx_Ctl_1	PHY1	O	Reserved for future use
P106	RGMII_TxC_1	PHY1	O	Reserved for future use
P107	RGMII_RxD_1(3:0)	PHY1	I	Reserved for future use
P108	RGMII_Rx_Ctl_1	PHY1	I	Reserved for future use
P109	RGMII_RxC_0	PHY1	I	Reserved for future use
EMAC 1 MDIO Interface				
P110	MdC_1	PHY1	O	MII Management Clock
P111	MdIO_1	PHY1	I/O	MII Management Data
P112	EMAC1ClientAnInterrupt	V4EMACDST1	O	Auto negotiation complete interrupt
Host Interface				
P113	HostOpCode(1:0)	V4EMACDST0	I	Host read / write indication
P114	HostReq	V4EMACDST0	I	Host IF request
P115	HostMiimSel	V4EMACDST0	I	Host select for MII management
P116	HostAddr(9:0)	V4EMACDST0	I	Host address
P117	HostWrData(31:0)	V4EMACDST0	I	Data written through Host IF
P118	HostMiimRdy	V4EMACDST0	O	MII management / host IF is ready
P119	HostRdData(31:0)	V4EMACDST0	O	Data read through Host IF
P120	HostEmac1Sel	V4EMACDST0	I	Host select for EMAC no. 1
P121	HostClk	V4EMACDST0	I	Host clock (1 - 100 MHz)
SGMII MGT Clocks				
P122	MGTClk_P	CLK	I	MGT clock positive
P123	MGTClk_N	CLK	I	MGT clock negative

Table 2: **HARD_TEMAC I/O Signals *(Cont'd)***

Port	Signal Name	Interface	I/O	Description
Reset				
P124	Reset	RST	I	Reset
Reference Clock				
P125	RefClk	CLK	I	Reference Clock

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.

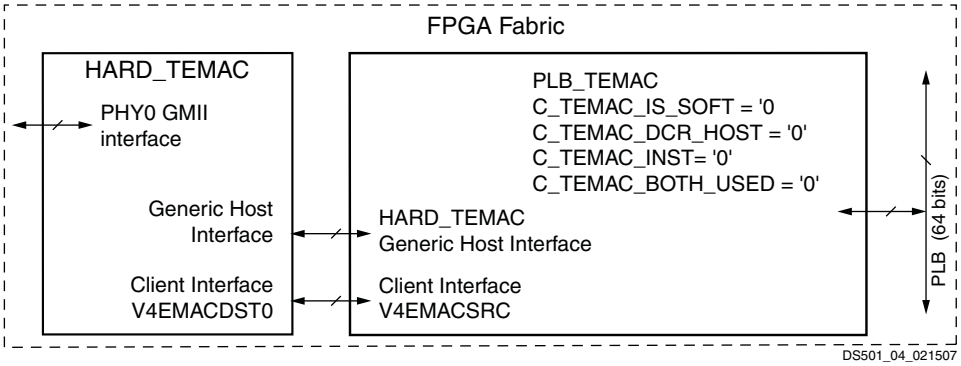

Figure 4: **System with One PLB_TEMAC**

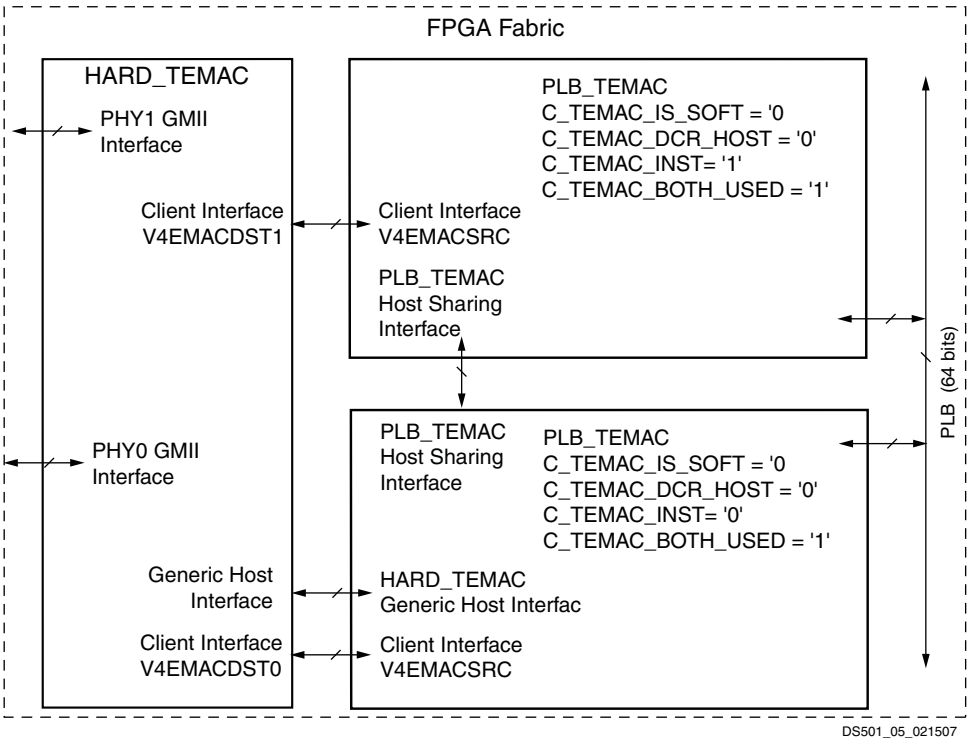

Figure 5: **System with Two PLB_TEMACs**

HARD_TEMAC Registers Definition

The HARD_TEMAC core registers are listed in Table 3. 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.

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 parameter settings of the hard_temac core such as which PHY interface is to be used. Where this is the case, the variable reset value will be identified.

Table 3: **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

HARD_TEMAC Core Registers

Note: Receiver Configuration Word 0 (RXC0)

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

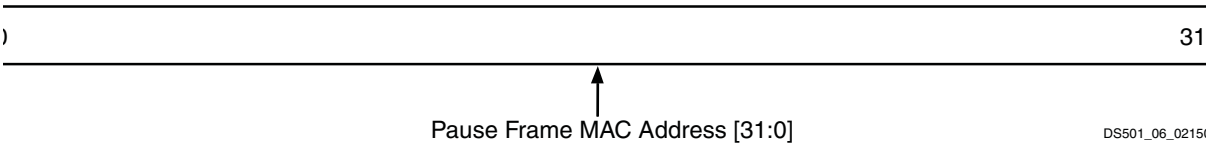

Figure 6: **Receiver Configuration Word 0 (offset 0x3200)**

Table 4: **Receiver Configuration Word 0 Register Bit Definitions**

Bit	Name	Core Access	Reset Value	Description
0 - 31	ERXC0	Read/Write	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.

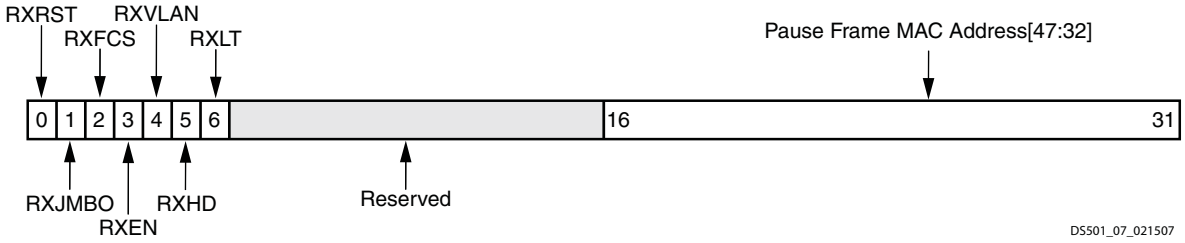

Figure 7: **Receiver Configuration Word 1 (offset 3240)**

Table 5: **Receiver Configuration Word 1 Register Bit Definitions**

Bit	Name	Core Access	Reset Value	Description
0	RXRST	Read/Write	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	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	1	**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.
3	RXEN	Read/Write	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	1	**VLAN enable**: When set to “1” VLAN tagged frames will be accepted by the receiver.
5	RXHD	Read/Write	0	**Half Duplex**: If “1”, the receiver will operate in half duplex mode. **Half Duplex is not supported so this bit should always be ’0’.**
6	RXLT	Read/Write	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	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.

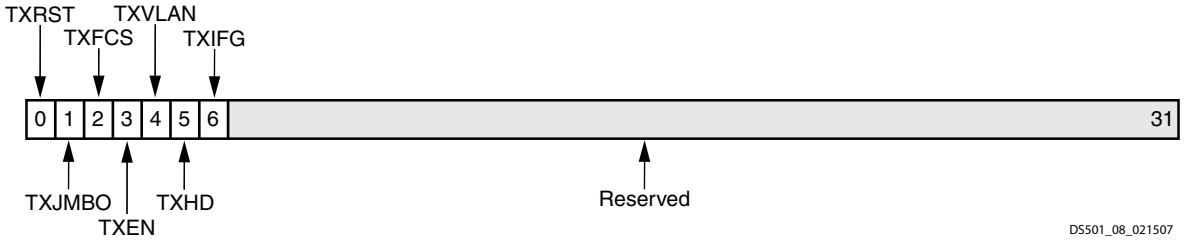

Figure 8: **Transmitter Configuration (offset 0x3280)**

Table 6: **Transmitter Configuration Register Bit Definitions**

Bit	Name	Core Access	Reset Value	Description
0	TXRST	Read/Write	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	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	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	1	**Transmitter enable**: If set to “1” the HARD_TEMAC transmitter is enabled. Otherwise, the core transmitter is disabled.
4	TXVLAN	Read/Write	1	**VLAN enable**: When set to “1” VLAN tagged frames will be sent by the transmitter.
5	TXHD	Read/Write	0	**Half Duplex**: If “1”, the transmitter will operate in half duplex mode. **Half Duplex is not supported so this bit should always be ’0’.**
6	TXIFG	Read/Write	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.

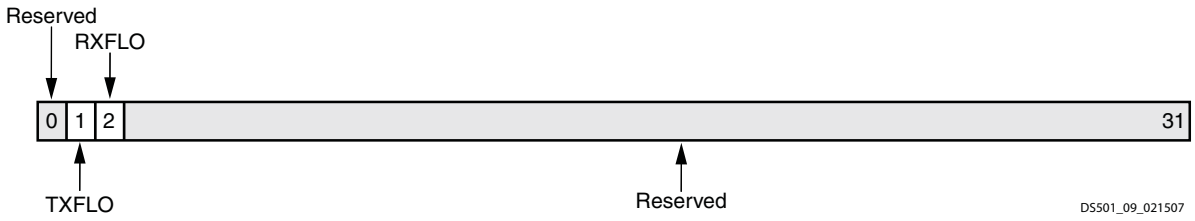

Figure 9: **Flow Control Configuration Register (offset 0x3232C0)**

Table 7: **Flow Control Configuration Register Bit Definitions**

Bit	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	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	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.

This release supports the MII, GMII, and SGMII interfaces. It does NOT support the RGMII or 1000BaseX interfaces.

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.

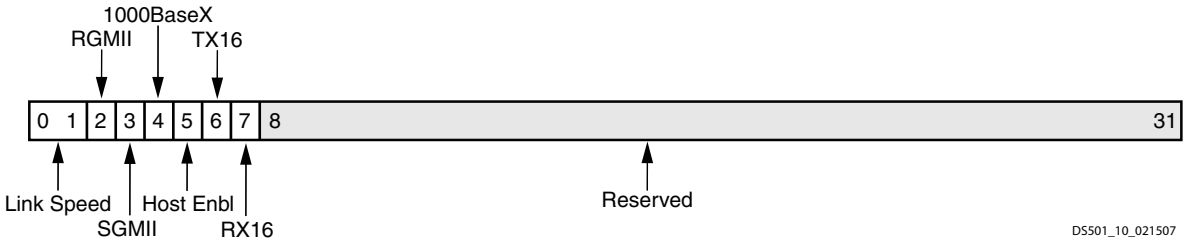

Figure 10: **EMAC Mode Configuration Register (offset 0x3300)**

Table 8: **EMAC Mode Configuration Register Bit Definitions**

Bit	Name	Core Access	Reset Value	Description
0 - 1	Link Speed	Read/Write	10 or 01 based on C_PHY_TYPE	**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	0	**RGMII mode enable**: When set to “1”, RGMII is enabled.
3	SGMII	Read	0 or 1 based on C_PHY_TYPE	**SGMII mode enable**: When set to “1”, SGMII is enabled.
4	1000 BaseX	Read	0	**1000BaseX mode enable**: When this bit is set to “1”, the Ethernet MAC is configured in 1000Base-X mode.
5	Host Enable	Read	1	**Host Interface Enable**: When this bit is set to “1”, the host interface is used.
6	TX 16 Bit	Read	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	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.

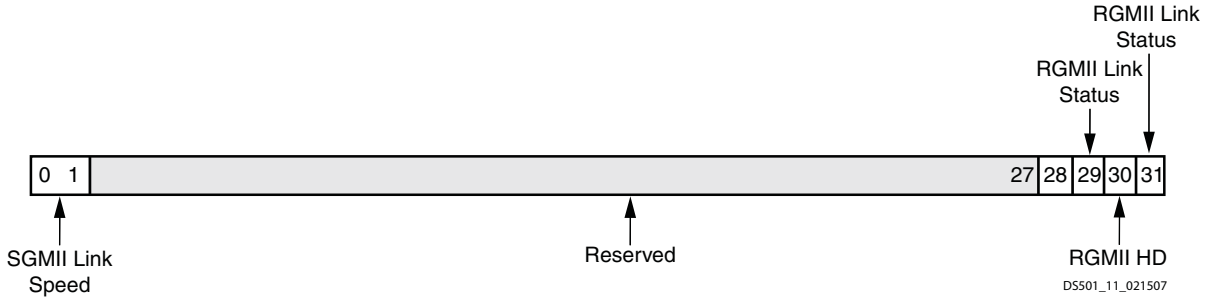

Figure 11: **RGMII/SGMII Configuration Register (offset 0x3320)**

Table 9: **RGMII/SGMII Configuration Register Bit Definitions**

Bit	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.
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 PLB_TEMAC specification provides a simple example of C code that demonstrates writing and reading of PHY registers using the MII Management interface.

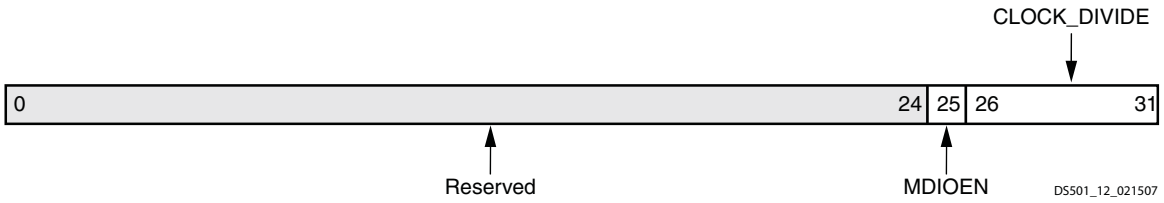

Figure 12: **Management Configuration Register (offset 0x3340)**

Table 10: **Management Configuration Register Bit Definitions**

Bit	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	1	**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.

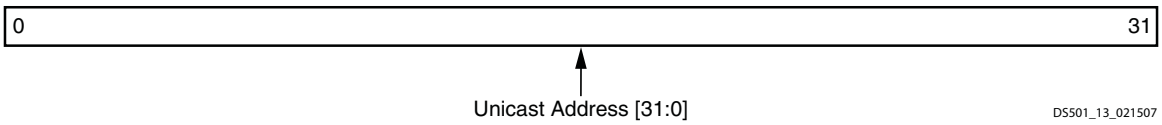

Figure 13: **Unicast Address Register Word 0 (offset 0x3380)**

Table 11: **Unicast Address Register Word 0 Bit Definitions**

Bit	Name	Core Access	Reset Value	Description
0 - 31	UAW0	Read/Write	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.

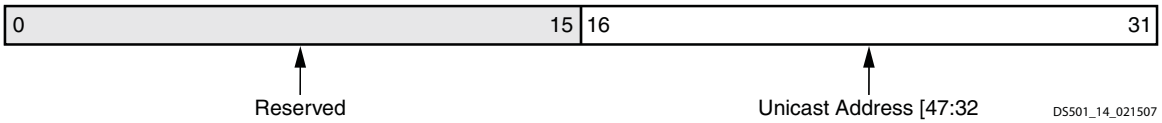

Figure 14: **Unicast Address Register Word 1 (offset 0x3384)**

Table 12: **Unicast Address Register Word 1 Bit Definitions**

Bit	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	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 PLB_TEMAC specification provides a simple example of C code that demonstrates writing and reading the multicast address CAM.

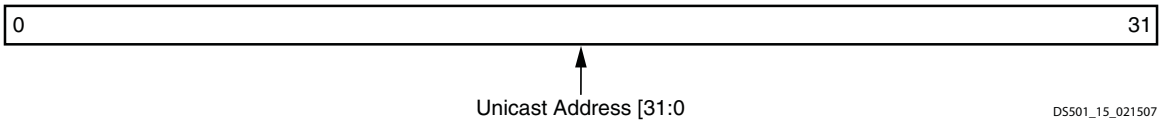

Figure 15: **Multicast Address Register Word 0 (offset 0x3388)**

Table 13: **Unicast Address Register Word 1 Bit Definitions**

Bit	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	0x0	**MAC Unicast Address bits [47:32]**.

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 PLB_TEMAC specification provides a simple example of C code that demonstrates writing and reading the multicast address CAM.

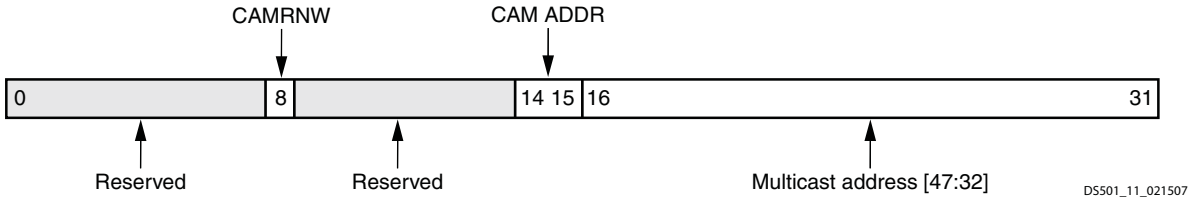

Figure 15: **Multicast Address Register Word 1 (offset 0x338C)**

Table 14: **Multicast Address Register Word 1 Bit Definitions**

Bit	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.

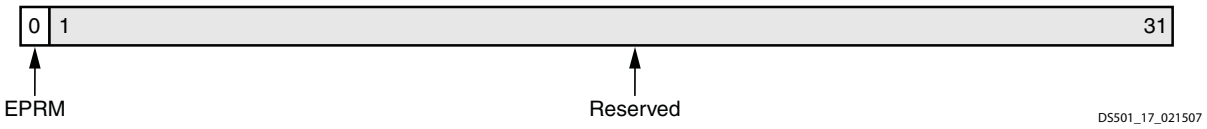

Figure 16: **Address Filter Mode Register (offset 0x3390)**

Table 15: **Address Filter Mode Register Bit Definitions**

Bit	Name	Core Access	Reset Value	Description
0	EPRM	Read / Write	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.

Design Implementation

Design Tools

The HARD_TEMAC design is implemented using VHDL code.

To see the synthesis tool used for this device, go to Tools. The NGC netlist output is then input to the Xilinx Foundation tool suite for device implementation.

Target Technology

The target technology is an FPGA listed in EDK Supported Device Families.

Device Utilization and Performance Benchmarks

Since the HARD_TEMAC is a VHDL wrapper around a hard silicon component, this core does not utilize FPGA fabric resources.

Specification Exceptions

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

Support

Xilinx provides technical support for this LogiCORE product when used as described in the product documentation. Xilinx cannot guarantee timing, functionality, or support of product if implemented in devices that are not defined in the documentation, if customized beyond that allowed in the product documentation, or if changes are made to any section of the design labeled *DO NOT MODIFY*.

Reference Documents

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

1. IEEE Std. 802.3-2000
2. EDK Processor IP Reference Guide
3. UG070 Virtex-4 FPGA User Guide
4. UG074 Virtex-4 FPGA Embedded Tri-Mode Ethernet MAC User Guide
5. DS416 Direct Memory Access and Scatter Gather v2.01a Data Sheet
6. DS459 PLB Master LocalLink v3.02a Data Sheet
7. DS458 PLB IPIF v1.00f Data Sheet
8. DS489 PLB_TEMAC (v3.00a) Data Sheet

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
06/22/05	1.0	Initial Xilinx release for version 1.00.a
8/4/05	1.1	Converted to new DS template.
11/18/05	2.0	Update for core revision 3.00.a
03/15/06	3.0	Update for latest revision 3.00.a information
07/07/06	3.1	Add two new parameters for RGMII delay constraints
09/14/06	3.2	New version to match new core version
2/15/07	3.3	Updated images to graphic standards, updated legal footer to year 2007.
1/03/08	3.4	Added details about IDELAYCTRL use
4/24/09	3.5	Replaced references to supported device families and tool name(s) with hyperlink to PDF file.

Notice of Disclaimer