Ethernet IP Core

Author: Igor Mohor
igorm@opencores.org

Rev. 1.12
March 15, 2002

Revision History

Rev.	Date	Author	Description
0.1	13/03/01	Igor Mohor	First Draft
0.2	17/03/01	Igor Mohor	MDC clock divider changed. Instead of the clock select bits CLKS[2:0] the clock divider bits CLKDIV[7:0] are used.
1.0	21/03/01	Igor Mohor	MII module completed. Revision changed to 1.0 due to cvs demands.
1.1	16/04/01	IM	DMA support and buffer descriptors added.
1.2	24/05/01	IM	Registers revised.
1.3	05/06/01	IM	Status is written to the status registers. DMA channels 2 and 3 are not used any more. Figures that are implementation specific removed from the document.
1.4	03/07/01	IM	COLLCONF register changed bit width. BCKPRESS and BCKPNBEN bit removed from MODER. LOOPBCK added.
1.5	21/07/01	IM	Signal RD0_O (Restart Descriptor for channel 0) added. Per packet CRC, BD changed.
1.6	03/12/01	IM	BD section rewritten.
1.7	05/12/01	IM	TX_BD_NUM register used instead of RX_BD_BASE_ADDR register.
1.8	07/01/02	IM	Minor typos fixed
1.9	30/01/02	IM	RST bit in MODER register is 1 after reset, initial collision window changed.
1.10	18/02/02	IM	Address recognition system added. Buffer Descriptors changed. DMA section changed. Ports changed.
1.11	02/03/02	IM	Typos fixed, INT_SOURCE and INT_MASK registers changed.
1.12	15/03/02	IM	TX_BD_NUM, MAC_ADDR0 and MAC_ADDR1 register description changed.

Contents:

1 Introduction

Ethernet IP Core consists of several modules (transmit, receive and control module forming a MAC module, MII Management module and the Host Interface) and is capable of operating at 10 or 100 Mbit/s for Ethernet and Fast Ethernet applications. For the complete Ethernet solution an external PHY is needed.

2

IO ports

This section describes the Ethernet Core IO ports

2.1 Ethernet Core IO ports

Ethernet core is connected to the "world" by three types of signals:

- Host interface (connection to the HOST (CPU) by the Wishbone)
    - Commons ports
    - Ports when Internal DMA is used
    - Ports when External DMA is used
- PHY Interface (connection to the PHY by the MII Management signals)
- Reset signals (for resetting different parts of the Ethernet core)

Different signals are used for the host interface when internal or external DMA is used.

2.1.1 Host Interface ports (common ports)

The below table contains the Ethernet IP Core to Host Interface common ports. The host interface is WISHBONE Rev. B compliant.

WISHBONE
COMPATIBLE

All signals listed below are active HIGH, unless otherwise noted. Signal direction is in respect to the Ethernet IP Core.

Port	Width	Direction	Description
CLK_I	1	I	Clock input
RST_I	1	I	Reset Input
ADDR_I	32	I	Address Input

DATA_I	32	I	Data Input
DATA_O	32	O	Data Output
SEL_I	4	I	Select Input Array Indicates which bytes are valid on data bus. Whenever this signal is not 1111b during a valid access, the ERR_O is asserted.
WE_I	1	I	Write Input Indicates a Write Cycle when asserted high or Read Cycle when asserted low.
STB_I	1	I	Strobe Input Indicates the beginning of a valid transfer cycle.
CYC_I	1	I	Cycle Input Indicates that a valid bus cycle is in progress.
ACK_O	1	O	Acknowledgment Output Indicates a normal Cycle termination.
ERR_O	1	O	Error Acknowledgment Output Indicates an abnormal cycle termination.
INTA_O	1	O	Interrupt Output A.

Table 1: Host Interface Ports (common ports)

2.1.2 Host Interface ports (internal DMA ports)

The below table contains the Ethernet IP Core to Host Interface ports when internal DMA is selected. The host interface is WISHBONE Rev. B compliant.

WISHBONE COMPATIBLE

All signals listed below are active HIGH, unless otherwise noted. Signal direction is in respect to the Ethernet IP Core.

Port	Width	Direction	Description
M_ADDR_O	32	O	Address Output
M_DATA_I	32	I	Data Input
M_DATA_O	32	O	Data Output
M_SEL_O	4	I	Select Output Array Indicates which bytes are valid on data bus. Whenever this

			signal is not 1111b during a valid access, the ERR_I is asserted.
M_WE_O	1	O	Write Output Indicates a Write Cycle when asserted high or Read Cycle when asserted low.
M_STB_O	1	O	Strobe Output Indicates the beginning of a valid transfer cycle.
M_CYC_O	1	O	Cycle Output Indicates that a valid bus cycle is in progress.
M_ACK_I	1	I	Acknowledgment Input Indicates a normal Cycle termination.
M_ERR_I	1	I	Error Acknowledgment Input Indicates an abnormal cycle termination.

Table 2: Host Interface Ports (internal DMA used)

2.1.3 Host Interface ports (external DMA ports)

The below table contains the Ethernet IP Core to Host Interface ports when external DMA is used. The host interface is WISHBONE Rev. B compliant.

WISHBONE COMPATIBLE

All signals listed below are active HIGH, unless otherwise noted. Signal direction is in respect to the Ethernet IP Core.

Port	Width	Direction	Description
REQ0_O	1	O	External DMA request to channel 0.
REQ1_O	1	O	External DMA request to channel 1.
ACK0_I	1	I	External DMA acknowledgment channel 0.
ACK1_I	1	I	External DMA acknowledgment channel 1.
ND0_O	1	O	Force Next Descriptor advancing for channel 0.
ND1_O	1	O	Force Next Descriptor advancing for channel 1.
RD0_O	1	O	Restart Descriptor for channel 0.

Table 3: Host Interface Ports (external DMA is used)

2.1.4 PHY Interface ports

The below table contains the Ethernet IP Core to PHY Interface ports. All signals listed below are active HIGH, unless otherwise noted. Signal direction is in respect to the Ethernet IP Core.

Port	Width	Direction	Description
MTxClk	1	I	**Transmit Nibble or Symbol Clock**. The PHY provides the MTxClk signal. It operates at the frequency of 25 MHz (100 Mbit/s) or 2.5 MHz (10 Mbit/s). The clock is used as a timing reference for the transfer of MTxD[3:0], MTxEn and MTxErr.
MTxD[3:0]	4	O	**Transmit Data Nibble**. Signals are the transmit data nibble. They are synchronized to the rising edge of MTxClk. When MTxEn is asserted, PHY accepts the MTxD.
MTxEn	1	O	**Transmit Enable**. When asserted, signal indicates to the PHY that the data MTxD[3:0] is valid and the transmission can start. The transmission starts with the first nibble of the preamble. Signal remains asserted until all nibbles to be transmitted are presented to the PHY. It is deasserted prior to the first MTxClk following the final nibble of a frame.
MTxErr	1	O	**Transmit Coding Error**. When asserted for one MTxClk clock period while MTxEn is also asserted, it causes the PHY to transmit one or more symbols that are not part of the valid data or delimiter set somewhere in the frame being transmitted to indicate that there has been a transmit coding error.
MRxClk	1	I	**Receive Nibble or Symbol Clock**. The PHY provides the MRxClk signal. It operates at the frequency of 25 MHz (100 Mbit/s) or 2.5 MHz (10 Mbit/s). The clock is used as a timing reference for the reception of MRxD[3:0], MRxDV and MRxErr.
MRxDV	1	I	**Receive Data Valid**. The PHY asserts this signal to indicate to the Rx MAC that it is presenting the valid nibbles on the MRxD[3:0] signals. Signal is asserted synchronously to the MRxClk. MRxDV is asserted from the first recovered nibble of the frame to the final recovered nibble. It is then deasserted prior to the first MRxClk that follows the final nibble.
MRxD [3:0]	4	I	**Receive Data Nibble**. Signals are the receive data nibble. They are synchronized to the rising edge of MRxClk. When MRxDV is asserted, the PHY sends a data nibble to the Rx MAC. For a correctly interpreted frame, seven bytes of a preamble and a completely formed SFD must be passed across the interface.

MRxErr	1	I	**Receive Error**. PHY asserts this signal to indicate to the Rx MAC that a media error was detected during the transmission of the current frame. MRxErr is synchronous to the MRxClk and is asserted for one or more MRxClk clock periods and then deasserted.
MColl	1	I	**Collision Detected**. The PHY asynchronously asserts the collision signal MColl after the collision is detected on the media. When deasserted, no collision is detected on the media.
MCrS	1	I	**Carrier Sense**. The PHY asynchronously asserts the carrier sense MCrS signal after the medium is detected in a non-idle state. When deasserted, signal indicates that the media is in the idle state (and the transmission can start).
MDC	1	O	**Management Data Clock**. Clock for the MDIO serial data channel.
MDIO	1	I/O	**Management Data Input/Output**. Bi-directional serial data channel for PHY/STA communication.

Table 4: PHY Interface Ports

2.1.5 Reset Signals

MAC sub-modules can be reset by one or more separate signals. To reset the PHY, its RESET signal can be used. The reset signal might be asserted by the boars system control register or by writing an appropriate bit in a PHY register that causes the PHY reset.

3

Registers

This section describes all base, control and status register inside the Ethernet IP Core. The Address field indicates a relative address in hexadecimal. Width specifies the number of bits in the register, and Access specifies the valid access types to that register. Where RW stands for read and write access, RO for read only access. A ‘C’ appended to RW or RO, indicates that some or all of the bits are cleared after a read.

Name	Address	Width	Access	Description
MODER	0x00	32	RW	Mode Register
INT_SOURCE	0x04	32	RW	Interrupt Source Register
INT_MASK	0x08	32	RW	Interrupt Mask Register
IPGT	0x0C	32	RW	Back to Back Inter Packet Gap Register
IPGR1	0x10	32	RW	Non Back to Back Inter Packet Gap Register 1
IPGR2	0x14	32	RW	Non Back to Back Inter Packet Gap Register 2
PACKETLEN	0x18	32	RW	Packet Length (minimum and maximum) register
COLLCONF	0x1C	32	RW	Collision and Retry Configuration
RX_BD_NUM	0x20	8	RW	Receive buffer descriptor number
CTRLMODER	0x24	32	RW	Control Module Mode Register
MIIMODER	0x28	32	RW	MII Mode Register
MIICOMMAND	0x2C	32	RW	MII Commend Register
MIIADDRESS	0x30	32	RW	MII Address Register Contains the PHY address and the register within the PHY address.
MIITX_DATA	0x34	32	RW	MII Transmit Data The data to be transmitted to the PHY.
MIIRX_DATA	0x38	32	RW	MII Receive Data The data received from the PHY.
MIISTATUS	0x3C	32	RW	MII Status Register
MAC_ADDR0	0x40	32	RW	MAC Individual Address0 The LSB four bytes of the individual address are written to this register.
MAC_ADDR1	0x44	32	RW	MAC Individual Address1 The MSB two bytes of the individual address are written to this register.
ETH_HASH0_ADR	0x48	32	RW	HASH0 Register
ETH_HASH1_ADR	0x4C	32	RW	HASH1 Register

Table 5: Register List

3.1 MODER (Mode Register)

Bit #	Access	Description
31-17		Reserved
16	RW	RECSMALL - Receive Small Packets 0 = Packets smaller than MINFL are ignored. 1 = Packets smaller than MINFL are accepted.
15	RW	PAD – Padding enabled 0 = Do not add pads to short frames 1 = Add pads to short frames (until the minimum frame length is equal to MINFL).
14	RW	HUGEN – Huge Packets Enable 0 = The maximum frame length is MAXFL. Bytes after that are discarded. 1 = Frames up 64 kB are transmitted.
13	RW	CRCEN – CRC Enable 0 = Tx MAC does not append the CRC (passed frames already contain the CRC. 1 = Tx MAC appends the CRC to every frame.
12	RW	DLYCRCEN – Delayed CRC Enabled 0 = Normal operation (CRC calculation starts immediately after the SFD). 1 = CRC calculation starts 4 bytes after the SFD.
11	RW	RST – Reset MAC 0 = Normal operation. 1 = MAC is reset.
10	RW	FULLD – Full Duplex 0 = Half duplex mode. 1 = Full duplex mode.
9	RW	EXDFREN – Excess Defer Enabled 0 = When the excessive deferral limit reached a packet is aborted. 1 = MAC waits for the carrier indefinitely.
8	RW	NOBCKOF – No Backoff 0 = Normal operation (a binary exponential backoff algorithm is used). 1 = Tx MAC starts retransmitting immediately after the collision.
7	RW	LOOPBCK – Loop Back 0 = Normal operation. 1 = TX is looped back to the RX.
6	RW	IFG – Interframe Gap for Incoming frames 0 = Normal operation (Minimum IFG is required for a frame to be accepted).

		1 = All frames are accepted regardless to the IFG.
5	RW	PRO – Promiscuous 0 = Check the destination address of the incoming frames. 1 = Receive the frame regardless of its address.
4	RW	IAM – Individual Address Mode 0 = Normal operation (Physical address is checked when the frame is received. 1 = The individual hash table is used to check all individual addresses that are received.
3	RW	BRO – Broadcast Address 0 = Receive all frames containing the broadcast address 1 = Reject all frames containing the broadcast address unless the PRO bit = 1.
2	RW	NOPRE – No Preamble 0 = Normal operation (7-byte preamble) 1 = No preamble is send
1	RW	TXEN – Transmit Enable 0 = Transmit is disabled 1 = Transmit is enabled
0	RW	RXEN – Receive Enable 0 = Receive is disabled 1 = Receive is enabled

Table 6: MODER Register

Reset Value:

MODER: 0000A800h

3.2 INT_SOURCE (Interrupt Source Register)

Bit #	Access	Description
31-7		Reserved
6	RW	RXC – Receive Control Frame Bit indicates that control frame was received. Bit is cleared by writing 1 to it.
5	RW	TXC – Transmit Control Frame Bit indicates that a control frame was transmitted. Bit is cleared by writing 1 to it.
4	RW	BUSY - Busy Bit indicates that a buffer was received and discarded due to a lack of buffers. Bit is cleared by writing 1 to it.
3	RW	RXE - Receive Error

		Bit indicates that an error occurred while receiving data. Bit is cleared by writing 1 to it.
2	RW	RXB - Receive Frame Bit indicates that a frame was received. Bit is cleared by writing 1 to it.
1	RW	TXE - Transmit Error Bit indicates that a buffer was not transmitted due to a transmit error. Bit is cleared by writing 1 to it.
0	RW	TXB - Transmit Buffer Bit indicates that a buffer has been transmitted. Bit is cleared by writing 1 to it.

Table 7: INT_SOURCE Register

Reset Value:
INT_SOURCE: 00000000h

3.3 INT_MASK (Interrupt Mask Register)

Bit #	Access	Description
31-7		Reserved
6	RW	RXC_M – Receive Control Frame Mask 0 = Event masked. 1 = Event causes an interrupt.
5	RW	TXC_M – Transmit Control Frame Mask 0 = Event masked. 1 = Event causes an interrupt.
4	RW	BUSY_M - Busy Mask 0 = Event masked. 1 = Event causes an interrupt.
3	RW	RXE_M – Receive Error Mask 0 = Event masked. 1 = Event causes an interrupt.
2	RW	RXF_M – Receive Frame Mask 0 = Event masked. 1 = Event causes an interrupt.
1	RW	TXE_M - Transmit Error Mask 0 = Event masked. 1 = Event causes an interrupt.
0	RW	TXB_M - Transmit Buffer Mask 0 = Event masked. 1 = Event causes an interrupt.

Table 8: INT_MASK Register

Reset Value:

INT_MASK: 00000000h

3.4 IPGT (Back to Back Inter Packet Gap Register)

Bit #	Access	Description
31-7		Reserved
6-0	RW	IPGT - Back to Back Inter Packet Gap Full Duplex: The recommended value is 0x15, which equals to 0.96 μs IPG (100 Mbit/s) or 9.6 μs (10 Mbit/s). The desired period in nibble times minus 6 should be written to the register. Half Duplex: The recommended value and default is 0x12, which equals to 0.96 μs IPG (100 Mbit/s) or 9.6 μs (10 Mbit/s). The desired period in nibble times minus 3 should be written to the register.

Table 9: IPGT Register

Reset Value:

IPGT: 00000012h

3.5 IPGR1 (Non Back to Back Inter Packet Gap Register 1)

Bit #	Access	Description
31-7		Reserved
6-0	RW	IPGR1 - Non Back to Back Inter Packet Gap 1 When carrier sense appears within the IPGR1 window, Tx MAC defers and the IPGR counter is reset. When carrier sense appears later then the IPGR1 window, the IPGR counter continuous counting. The recommended and default value for this register is 0xC and must be within the range [0,IPGR2].

Table 10: IPGR1 Register

Reset Value:

IPGR1: 0000000Ch

3.6 IPGR2 (Non Back to Back Inter Packet Gap Register 2)

Bit #	Access	Description
31-7		Reserved
6-0	RW	IPGR2 - Non Back to Back Inter Packet Gap 2 The recommended and default value is 0x12, which equals to 0.96 µs IPG (100 Mbit/s) or 9.6 µs (10 Mbit/s).

Table 11: IPGR2 Register

Reset Value:

IPGR2: 00000012h

3.7 PACKETLEN (Packet Length Register)

Bit #	Access	Description
31-16	RW	MINFL - Minimum Frame Length Minimum Ethernet packet is 60 bytes long. If a reception of smaller frames is needed, assert the RECSMALL bit (in the mode register MODER) or change the value of this register. To transmit small packets, assert the PAD bit or the MINFL value. (See the PAD bit description in the MODER register).
15-0	RW	MAXFL - Maximum Frame Length Maximum Ethernet packet is 1518 bytes long. To support this and to leave some additional space for the tags, a default maximum packet length equals to 1536 bytes (0x0600). If there is a need to support bigger packets, you can assert the HUGEN bit or increase the value of the MAXFL field. (See the HUGEN bit description in the MODER).

Table 12: PACKETLEN Register

Reset Value:

PACKETLEN: 003C0600h

3.8 COLLCONF (Collision and Retry Configuration Register)

Bit #	Access	Description

31-20		Reserved
19-16	RW	MAXRET - Maximum Retry This field specifies the maximum number of consequential retransmission attempts after the collision is detected. When the maximum number is reached the Tx MAC reports an error and stops transmitting the current packet. According to the Ethernet standard, the MAXRET default value is set to 0xf (15).
15-6		Reserved
5-0	RW	COLLVALID - Collision Valid This field specifies a collision time window. Collision that occurs later than the time window is reported as a »Late Collisions« and transmission of the current packet is aborted. The default value equals to 0x3f (By default late collision is every collision that occurs 64 bytes (63 + 1) from the preamble).

Table 13: COLLCONF Register

Reset Value:

COLLCONF: 000F003fh

3.9 TX_BD_NUM (Transmit BD Number Reg.)

Bit #	Access	Description
31:8		Reserved
9:0	RW	Transmit Buffer Descriptor (Tx BD) Number Number of the Tx BD (when external DMA is used) or number of the Tx BD multiplied by 2 (when internal DMA is used).

Table 14: TX_BD_NUM Register

Reset Value:

TX_BD_NUM: 00000080h

3.10 CTRLMODER (Control Module Mode Register)

Bit #	Access	Description
31-3		Reserved
2	RW	TXFLOW – Transmit Flow Control

		0 = PAUSE control frames are blocked. 1 = PAUSE control frames are allowed to be sent.
1	RW	RXFLOW – Receive Flow Control 0 = Received PAUSE control frames are ignored. 1 = Transmit function (Tx MAC) is blocked when a PAUSE control frame is received.
0	RW	PASSALL – Pass All Receive Frames 0 = Control frames are not passed to the host. MAC Control module is enabled. 1 = All received frames are passed to the host. MAC Control module is disabled.

Table 15: CTRLMODER Register

Reset Value:

CTRLMODER: 00000000h

3.11 MIIMODER (MII Mode Register)

Bit #	Access	Description
31-11		Reserved
10	RW	MIIMRST – Reset of the MIIM module
9		Reserved
8	RW	MIINOPRE – No Preamble 0 = 32-bit preamble send 1 = no preamble send
7-0	RW	CLKDIV – Clock Divider The field is a host clock divider factor. The host clock can be divided by an even number, greater then 1. The default value is 0x64 (100).

Table 16: MIIMODER Register

Reset Value:

MIIMODER: 00000064h

3.12 MIICOMMAND (MII Command Register)

Bit #	Access	Description
31-3		Reserved
2	RW	WCTRLDATA – Write Control Data
1	RW	RSTAT – Read Status
0	RW	SCANSTAT – Scan Status

Table 17: MIICOMMAND Register

Reset Value:
MIICOMMAND: 00000000h

3.13 MIIADDRESS (MII Address Register)

Bit #	Access	Description
31-13		Reserved
12-8	RW	RGAD – Register Address (within the PHY selected by the FIAD[4:0])
7-5		Reserved
4-0	RW	FIAD – PHY Address

Table 18: MIIADDRESS Register

Reset Value:
MIIADDRESS: 00000000h

3.14 MIITX_DATA (MII Transmit Data)

Bit #	Access	Description
31-16		Reserved
15-0	RW	CTRLDATA – Control Data (data to be written to the PHY)

Table 19: MIITX_DATA Register

Reset Value:
MIITX_DATA: 00000000h

3.15 MIIRX_DATA (MII Receive Data)

Bit #	Access	Description
31-16		Reserved
15-0	RW	PRSD – Received Data (data read from the PHY)

Table 20: MIIRX_DATA Register

Reset Value:

MIIRX_DATA: 00000000h

3.16 MIISTATUS (MII Status Register)

Bit #	Access	Description
31-3		Reserved
2	R	NVALID – Invalid 0 = The data in the MSTATUS register is valid. 1 = The data in the MSTATUS register is invalid.
1	R	BUSY 0 = The MII is ready. 1 = The MII is busy (operation in progress).
0	RC	LINKFAIL: 0 = Link OK 1 = Link Fail Link fail condition occurred (link might be OK now). After read this bit is cleared. Another status read gets new status.

Table 21: MIISTATUS Register

Reset Value:

MIISTATUS: 00000000h

3.17 MAC_ADDR0 (MAC Address Register 0)

Bit #	Access	Description
31-24	RW	Byte 2 of the Ethernet MAC address (individual address)

23-16	RW	Byte 3 of the Ethernet MAC address (individual address)
15-8	RW	Byte 4 of the Ethernet MAC address (individual address)
7-0	RW	Byte 5 of the Ethernet MAC address (individual address)

Table 22: MAC_ADDR0 Register

Reset Value:

MAC_ADDR0: 00000000h

Note: When address is transmitted, byte 0 is sent first and byte 5 is sent last.

3.18 MAC_ADDR1 (MAC Address Register 1)

Bit #	Access	Description
31-16		Reserved
15-8	RW	Byte 0 of the Ethernet MAC address (individual address)
7-0	RW	Byte 1 of the Ethernet MAC address (individual address)

Table 23: MAC_ADDR1 Register

Reset Value:

MAC_ADDR1: 00000000h

Note: When address is transmitted, byte 0 is first and byte 5 is last.

3.19 HASH0 (HASH Register 0)

Bit #	Access	Description
31-0	RW	Hash0 value

Table 24: HASH0 Register

Reset Value:

HASH0: 00000000h

3.20 HASH1 (HASH Register 1)

Bit #	Access	Description
31-0	RW	Hash1 value

Table 25: HASH1 Register

Reset Value:

HASH1: 00000000h

4

Operation

This section describes the Ethernet IP Core operation.

The core consists of five modules:

- Host interface connects the Ethernet Core to the rest of the system via the Wishbone (using DMA transfers). Registers are also part of the host interface.
- TX Ethernet MAC performs transmit function.
- RX Ethernet MAC performs receive function.
- MAC Control Module performs full duplex flow control function.
- MII Management Module performs PHY control and gathers the status information from it.

All modules together perform a full function 10/100 Mbit/s Media Access Control. The Ethernet core can operate in a half or a full duplex mode. The basic of the Ethernet is CSMA/CD protocol. The CSMA/CD stands for Carrier Sense Multiple Access / Collision Detection.

In half duplex mode when a station wants to transmit, it has to observe the activity on the media (Carrier Sense). As soon as the media is idle (no one is transmitting), any station can start with the transmission (Multiple Access). If two or more stations are transmitting at the same time, a collision on the media is detected. All stations stop transmitting and back-off for some random time. After the back-off time, the station checks the activity on the media again. If the media is idle, it starts transmitting. All other stations wait for the current transmission to end.

In full duplex mode the carrier sense and the collision detect signals are ignored. Flow control is achieved by sending and receiving the PAUSE control frames (see the TXFLOW and RXFLOW bit description in the CTRLMODER register). The MAC Control module takes care of that.

MII Management module provides a media independent interface (MII) to the external PHY. Using MIIM module, PHY’s configuration and status registers can be read/write.

4.1 Host Interface Operation

The host interface connects the Ethernet IP core to the rest of the system (RISC, memory) via the Wishbone bus. The Wishbone is used for accessing the configuration registers and the memory. At the moment only DMA transfers are supported for transferring the data from/to the memory.

4.1.1 Configuration Registers

The function of the configuration registers is transparent and can be easily understood by reading the Registers section (Section 3) on the page 11.

4.1.2 External / Internal DMA operation

The Ethernet MAC can use external or internal DMA engine:

- **external DMA:** External DMA core is needed (available from Opencores website). This mode is not recommended because it will soon be obsolete.
- **internal DMA:** Core uses internal DMA resources.

Selection between external and internal DMA is done in the eth_defines.v file.

4.1.3 Buffer Descriptors (BD)

The transmission and the reception processes are based on the descriptors. The Transmit Descriptors (TxD) are used for transmission while the Receive Descriptors (RxD) are used for reception.

Buffer descriptors when using internal DMA are 64-bits long. First 32 bits are reserved for length and status while last 32 bits contain the pointer to the associated buffer (where data is stored). Ethernet MAC core has internal RAM that can store up to 128 BD’s (for both Rx and Tx).

Buffer descriptors when using external DMA are 32-bits long. Status and length information is stored to these 32 bits. Pointers to the associated buffers are stored in the external DMA core (additional descriptors that are part of the DMA). Ethernet MAC core has internal RAM that can store up to 256 BD’s (for both Rx and Tx).

First 32 bits are the same for both, internal and external DMA operation. The difference is in the pointer to the buffer (read the above section).

All descriptors are saved in the internal memory at addresses from 0x400 to 0x800 (128 64-bit descriptors when internal DMA is used or 256 32-bit descriptors when external DMA is used). Transmit descriptors are located between the start address (0x400) and the address that equals to the value written in the TX_BD_NUM register (page 17) multiplied by 4. This register holds the number of the Tx buffer descriptors used multiplied by 2. Receive descriptors are located between the start address (0x400) plus address number written in the TX_BD_NUM multiplied by 4 and the descriptor end address (0x800).

Transmit and receive status of the packet is written to the associated buffer descriptor once its transmission/reception is finished.

4.1.3.1 Tx Buffer Descriptors

Transmit descriptors contain information about associated buffers (length, status). When internal DMA is selected, they also contain pointers to the buffers where the data is.

ADDR = Offset + 0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LEN															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RD	**IRQ**	**WR**	**PAD**	**CRC**	Reserved		**UR**	**RTRY[3:0]**				**RL**	**LC**	**DF**	**CS**

ADDR = Offset + 4

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TXPNT															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXPNT															

Figure 1: Tx Buffer Descriptor

Bit #	**Access**	**Description**
31-16	RW	LEN – Length Number of bytes associated with this BD that need to be transmitted.
15	RW	RD – Tx BD Ready 0 = The buffer associated with this buffer descriptor is not ready and you are free to manipulate with it. After the data from associated buffer is transmitted or after an error condition occurs, this bit is cleared to 0. 1 = The data buffer is ready for transmission or is currently being transmitted. You are not allowed to manipulate with this descriptor once this bit is set.
14	RW	IRQ – Interrupt Request Enable 0 = No interrupt is generated after the transmission. 1 = When data associated with this buffer descriptor is send, TXB or TXE interrupt will be asserted (See 3.2 INT_SOURCE (Interrupt Source Register) on page 13 for more details).
13	RW	WR – Wrap 0 = This buffer descriptor is not the last descriptor in the buffer descriptor table.

		1 = This buffer descriptor is the last descriptor in the buffer descriptor table. After this buffer descriptor is used, the first buffer descriptor in the table will be used again.
12	RW	PAD – Pad Enable 0 = No pads will be add at the end of short packets. 1 = Pads will be add to the end of short packets.
11	RW	CRC – CRC Enable 0 = CRC won’t be added at the end of the packet 1 = CRC will be add at the end of the packet.
10:9		Reserved
8	RW	UR – Underrun Underrun occurred while sending this buffer
7:4	RW	RTRY – Retry Count Number of retries before the frame was successfully sent.
3	RW	RL – Retransmission Limit Set when the transmitter fails (Retry Limit + 1) attempts to successfully transmit a message because of repeated collisions on the medium. Retry limit is set in the COLLCONF register on page 16.
2	RW	LC – Late Collision Late collision occurred while sending this buffer. Transmission is stopped and this bit is written. Late collision is defined in the COLLCONF register on page 16.
1	RW	DF – Defer Indication The frame was deferred before being sent successfully, that is, the transmitter had to wait for carrier sense before sending because the line was busy. This is not a collision indication collisions are indicated in RTRY.
0	RW	CS – Carrier Sense Lost Carrier sense lost. Set when carrier sense is lost during frame transmission. The Ethernet controller writes CS after it finishes sending the buffer.

Table 26: Tx Buffer Descriptor

Bit #	Access	Description
31-0	RW	TXPNT – Transmit Pointer Pointer to the buffer when associated frame is stored.

Table 27: Tx Buffer pointer when internal DMA is used

4.1.3.2 Rx Buffer Descriptors

Receive BD's contain information about the received frames (length, status). When internal DMA is selected, they also contain pointers to the buffers where the data is.

ADDR = Offset + 0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LEN															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E	**IRQ**	**WR**	Reserved					**M**	**OR**	**IS**	**DN**	**TL**	**SF**	**CRC**	**LC**

ADDR = Offset + 4

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RXPNT															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXPNT															

Figure 2: Rx Buffer Descriptor

Bit #	**Access**	**Description**
31-16	RW	LEN– Number of the received bytes, associated with this BD.
15	RW	E - Empty 0 = The data buffer associated with this buffer descriptor has been filled with data or has stopped because an error occurred. The core can read or write this BD. As long as this bit is zero, this buffer descriptor won't be used. 1 = The data buffer is empty (and ready for receiving data) or currently receiving data.
14	RW	IRQ – Interrupt Request Enable 0 = No interrupt is generated after the reception. 1 = When data is received (or error occurs), RXF interrupt will be asserted (See 3.2 INT_SOURCE (Interrupt Source Register) on page 13 for more details).
13	RW	WRAP 0 = This buffer descriptor is not the last descriptor in the buffer descriptor table. 1 = This buffer descriptor is the last descriptor in the buffer descriptor table. After this buffer descriptor is used, the first Rx buffer descriptor in the table will be used again.
12:8		Reserved.
7	RW	M – Miss

		0 = The frame is received because of an address recognition hit. 1 = The frame is received because of promiscuous mode. The Ethernet controller sets M for frames that are accepted in promiscuous mode, but are tagged as a miss by internal address recognition. Thus, in promiscuous mode, M determines whether a frame is destined for this station.
6	RW	OR – Overrun Set when a receiver overrun occurs during frame reception.
5	RW	IS – Invalid Symbol Set when reception of an invalid symbol is detected by the PHY.
4	RW	DN – Dribble Nibble Set when a received frame can not de divided by 8 (one extra nibble received).
3	RW	TL – Too Long Set when a received frame is too long (bigger than the value set in the PACKETLEN register (page 16).
2	RW	SF – Short Frame Set when a frame that is smaller than the minimum length is received (minimum length is set in the PACKETLEN register (page 16)).
1	RW	CRC – Rx CRC Error Set when a received frame contains a CRC error.
0	RW	LC – Late Collision Set when a late collision occurred while receiving frame.

Table 28: Rx Buffer Descriptor

Bit #	Access	Description
31-0	RW	RXPNT – Receive Pointer Pointer to the buffer when associated frame is stored.

Table 29: Rx Buffer pointer when internal DMA is used

4.1.4 Frame Transmission

There are few differences in the frame transmission when external or internal DMA are used.

4.1.4.1 Frame Transmission when Internal DMA is used

When RISC wants to transmit the first frame, it has to do several things:

- Store frame to the memory.
- Associate the Tx BD in the Ethernet MAC core with the packet written to the memory (length, pad, crc, etc.). See section 4.1.3 Buffer Descriptors (BD) on page 24.
- Enable the TX part of the Ethernet Core by setting the TXEN bit to 1.

As soon as Ethernet core is enabled, it continuously reads the first BD. As soon as the descriptor is marked as ready, core reads the pointer to the memory where the associated data is stored. Then it starts reading data to the internal FIFO. As soon as the FIFO is full, transmission begins.

At the end of the transmission transmit status is written to the buffer descriptor and interrupt might be generated (when enabled). There are two possibilities what will happen next (according to the WR bit (wrap) in the descriptor):

- If WR bit is not set, BD address is incremented, next descriptor is loaded and everything starts all over again (if next BD is marked as ready).
- If WR bit is set, first BD address (base) is loaded again. As soon as the BD is marked as ready, transmission will start.

4.1.4.1 Frame Transmission when External DMA is used

When RISC wants to transmit the first frame, it has to do several things:

- Store frame to the memory.
- Associate the first DMA descriptor with the stored frame:
    - Source address field points to the frame in the memory.
    - Destination field points to the Ethernet host interface.
    - Next pointer field points to the next descriptor in the memory (this descriptor is not stored, yet).
    - Total transfer size equals is equal to the stored frame size.
    - Increment source and destination address fields must be set to zero. We always go to the descriptor that is written in the "Next descriptor field" and use its source and destination addresses.
    - EOL bit must be set to zero. This means that the descriptor is the last descriptor in the list (and frame is the last frame).
    - Set the channel 0 registers (channel 0 is used for transmitting).
    - Enable the channel by setting CH_EN bit to 1. Channel is now enabled, and it waits for the REQ0 signal to be asserted, to start the operation.
- Associate the transmit buffer descriptor in the Ethernet MAC core with the packet written to the memory. Enable the TX part of the Ethernet Core by setting the TXEN and DMAEN bits to 1.

As soon as Ethernet core is enabled and the transmit descriptor is ready, it will assert the DMA request signal (REQ0). When DMA will notice the request, it will start transmitting the first word (32-bits) of the pointed frame. The REQ0 will be asserted each time 32-bits of the data will be needed. When last 32-bits will be needed the request for the last word the ND0 signal is asserted. The ND0 instructs the DMA to load the next descriptor after the last word is transmitted.

At the end of the transmission transmit status is written to the buffer descriptor and interrupt might be generated (is enabled). There are two possibilities what will happen next (according to the EOL bit in the descriptor):

- If EOL bit is not set, that means there are more frames stored in the memory waiting to be sent. New descriptor is loaded and everything starts all over again. Since DMA is already set, nothing needs to be set at the DMA core.
- If EOL bit is set, DMA channel will stop and clear the CH_EN bit in the channel CSR register. In order to start transmission everything needs to set like at the beginning of the transmission of the first packet.

4.1.5 Frame Reception

There are few differences in the frame reception when external or internal DMA are used.

4.1.5.1 Frame Reception when Internal DMA is used

When RISC wants to receive the first frame, it has to do several things:

- Set the receive buffer descriptor that will be associated with the received packet and mark it as empty.
- Enable the Ethernet receive function by setting the RECEN bit to 1.

Ethernet Core reads the Rx BD and if it is marked as empty, it's starts receiving frames. Ethernet receive function receives incoming frame nibble per nibble. After the whole frame is received and stored to the memory, receive status and pointer to the memory where the data is stored, are written to the BD. Interrupt might be generated (if enabled). Then BD address is incremented and next BD loaded. If new BD is marked as empty, another frame can be received, otherwise operation stops.

4.1.5.2 Frame Reception when External DMA is used

When RISC wants to receive the first frame, it has to do several things:

- Set the descriptors that will be used with the DMA channel 1 to appropriate values (where the incoming will be stored).
- Set the channel 1 to use descriptors and enable it
- Set the receive buffer descriptor that will be associated with the received packet.
- Enable the Ethernet receive function by setting the RECEN and DMAEN bits to 1.

Ethernet receive function receives incoming frame nibble per nibble. After it receives a whole word, it is written to the memory via the Wishbone by asserting the REQ1 signal (requesting the DMA write to the memory on the channel 1). After the whole frame is received, the ND1 signal is asserted to force usage of the next descriptor with the next frame. Receive status is written to the receive buffer descriptor.

4.2. DMA Operation

DMA operation allows for completely transparent data movement between the Ethernet core and the function attached to the WISHBONE bus. Once setup, no function micro controller intervention is needed for normal operations. The Ethernet core has two associated pairs of REQn and ACKn signals.

When the DMAEN bit in the MODER register is set, the Ethernet core will use the DMA_REQ and DMA_ACK signals for the DMA flow control. The DMA_REQ signal is asserted when there is data in the buffer (receiving frames) or when the buffer is empty and needs to be filled (transmitting frames). The DMA must reply with a DMA_ACK for each word (4 bytes) transferred. The buffer holds one MAX_PL_SZ packet (one word). Depending on DMA and external bus latency it could be set up to hold more than one packet. In that case additional action needs to be performed in the core (not supported, yet).

More information about Ethernet-DMA functionality can be found in the section 4.1.2 External / Internal DMA operation

The Ethernet MAC can use external or internal DMA engine:

- **external DMA:** External DMA core is needed (available from Opencores website). This mode is not recommended because it will soon be obsolete.
- **internal DMA:** Core uses internal DMA resources.

Selection between external and internal DMA is done in the eth_defines.v file.

4.1.3 Buffer Descriptors (BD) on page 24.

For more general information about DMA and descriptors, read the “WISHBONE DMA/Bridge Core specification” that is available on the Opencores web site.

4.3 TX Ethernet MAC

TX Ethernet MAC generates 10BASE-T/100BASE-TX transmit MII nibble data streams in response to the byte streams supplied from the transmit logic (host). It performs the required deferral and back-off algorithms, takes care of the inter packet gap (IPG), computes the checksum (FCS) and monitors the physical media (by monitoring Carrier Sense and collision signals). The TX Ethernet MAC is divided into several modules that provide the following functionality:

- Generation of the signals connected to Ethernet PHY that are used in transmission process
- Generation of the status signals that are used by the host for tracking the transmission process
- Random time generation, used in the back-off process after a collision is detected.
- CRC generation and checking
- Pad generation
- Data nibble generation

4.4 RX Ethernet MAC

RX Ethernet MAC transmits the data streams to the host in response to the 10BASE-T or 100BASE-TX received MII nibbles. The module is divided into several sub-modules that provide the following functionality:

- Preamble removal
- Data assembly (from input nibble to output byte)
- CRC checking for all incoming packets.
- Generation of the signal that can be used in the hash table for address recognition
- Generation of the status signals that are used by the host for tracking the reception process.

4.5 MAC Control Module

The MAC Control Module performs a real-time flow control function for the full duplex operation. Control opcode PAUSE is used for stopping the station that is transmitting the packets. The receive buffer (FIFO) starts to fill up when the upper layer cannot keep up accepting the incoming packets. Before an overflow happens, the upper layer sends a control frame PAUSE to the transmitting station. This control frame inhibits the transmission of the data frames for a specified period of time.

When the MAC Control module receives a PAUSE control frame, it loads the pause timer with the value received in the pause timer value field. The Tx MAC is stopped (paused) from transmitting the data frames for the “pause timer value” slot times. Pause timer decrements one each time a slot time passes by. When the pause time number equals to zero, the MAC transmitter resumes the transmit operation.

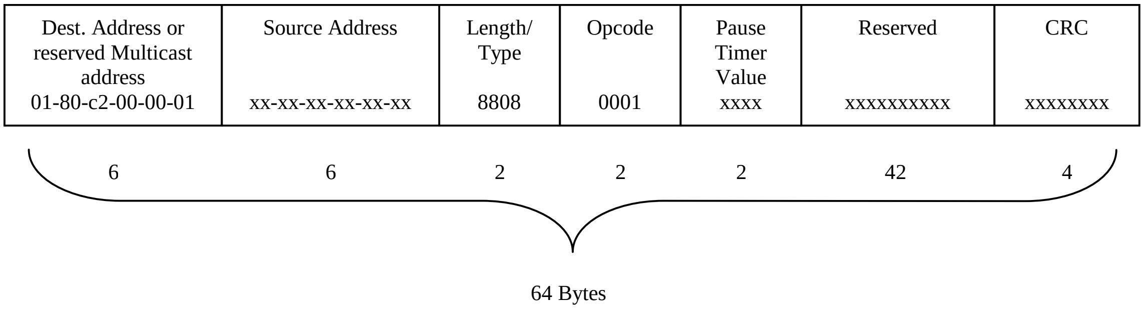
The MAC Control Module has the following functionality:

- Control frame detection
- Control frame generation
- TX/RX MAC Interface
- PAUSE Timer
- Slot Timer

4.5.1 Control Frame Detection

The incoming data packets are passed from the receiver through the MAC Control Module to the upper layers while control frames are usually dropped. The PASSALL bit in the CTRLMODER register defines whether the control frames are passed or dropped.

A valid PAUSE control frame has the frame structure described on Figure 3: Structure of the PAUSE control frame:



Dest. Address or reserved Multicast address	Source Address	Length/ Type	Opcode	Pause Timer Value	Reserved	CRC
01-80-c2-00-00-01	xx-xx-xx-xx-xx-xx	8808	0001	xxxx	xxxxxxxxxx	xxxxxxxx
6	6	2	2	2	42	4

Figure 3: Structure of the PAUSE control frame

Destination address must be a reserved multicast address (01-80-c2-00-00-01) or a destination address equal to the Ethernet IP Core MAC address. The Length/Type field must be equal to 8808 and the opcode must be equal to 0001 for a PAUSE control frame.

When the receive flow control and the MAC Control module are enabled (RXFLOW asserted and PASSALL deasserted), a PAUSE Timer Value from the PAUSE control frame is passed to the PAUSE timer.

4.5.2 Control Frame Generation

When the host wants to send a PAUSE control frame, it asserts the Transmit Pause Request (TPAUSERQ). When a request is detected, the control module waits for the current transmission to end. After that it starts transmitting the PAUSE control frame by asserting the Transmit Packet Start Frame (TxStartFrm), and by providing the appropriate control data. SendingCtrlFrm is used for instructing the Transmit function (TX Ethernet MAC) to pad and to append the FCS. The transmit Pause Frame End (TxEndFrm) is asserted at the end to inform the host that a Pause request was sent.

Asserting the TXFLOW bit in the MODER register enables the transmission of the PAUSE control frame.

The TPAUSERQ signal (request) is latched in the MAC Control Generator and reset after the PAUSE control frame is transmitted. This prevents issuing a new PAUSE request until the current request is send. The Transmit Pause Timer Value TPAUSETV[15:0] is set prior the transmit pause request is asserted. The TPAUSETV contains the value to be sent as a Pause Timer Value in the pause control frame (Figure 3: Structure of the PAUSE control frame).

4.5.3 TX/RX MAC Interface

The MAC Control Module is connected between the host and the Tx and Rx modules. When enabled, the MAC Control module logic takes over the control of the following signals: TxData[7:0], TxStartFrm, TxEndFrm, TxUsedData, TxDone and TxAbort. These signals are connected directly between the host and the MAC transmit and receive functions when data frames (not control frames) are transmitted or received.

On the other hand when a host wants to send a PAUSE control frame it asserts a request signal TPauseRQ and it is up to MAC Control Module to initiate the transmission. In this case the above signals are not connected to the host any more. The MAC Control module drives the appropriate control data signals and instructs the Tx module to transmit.

When a PAUSE control frame is received, the frame can be dropped or passed to the host, depending on the state of the PASSALL signal. Again TxData[7:0], TxStartFrm, TxEndFrm, TxUsedData, TxDone and TxAbort are not connected directly.

4.5.4 PAUSE Timer

The 16-bit PAUSE timer is loaded with a pause timer value, when a PAUSE control frame is received. The timer inhibits the data frame transmissions for the timer value time slots. This is done by:

- Preventing the Tx MAC module from seeing the signal TxStartFrm from the host.
- Preventing the host from seeing the signal TxUsedData from the Tx module.

The timer decrements one each time a time slot passes by. A Slot Timer is used for counting the slot time.

4.5.5 Slot Timer

Slot timer is activated when a PAUSE Timer is preloaded. It counts slot times and generates pulses to the PAUSE Timer for every slot time passed.

4.6 MII Management Module

MII Management module is a simple two-wire interface between the host and an external PHY device. It is used for configuration and status read of the physical device. The physical interface consists of a management data line MDIO and a clock line MDC. During the read/write operation the most significant bit is shifted in/out first from/to the MDIO data signal. On each rising edge of the MDC a shift register is shifted to the left and a new value appears on the MDIO.

Internally the interface consists of four signals:

- MDC (Management Data Clock)

- MDI (Management Data Input)
- MDO (Management Data Output)
- MDOEN (Management Data Output Enable)

The unidirectional lines MDI, MDO and MDOEN are combined together to make a bi-directional signal MDIO that is connected to the PHY.

The configuration and status data is written/read to/from the PHY through the MDIO signal.

Management Data Clock MDC is a low frequency clock, derived from the host clock by dividing it.

Three commands are supported for controlling the PHY:

- Write Control Data (Writes the control data to the PHY configuration registers)
- Read Status (Reads the PHY control and status register)
- Scan Status (Continuously reads the PHY status register of one or more PHYs (link fail status)).

The MII Management Module consists of four sub modules:

- Operation Controller
- Shift Registers
- Output Control Module
- Clock Generator

4.6.1 Operation Controller

Operation Controller's task is to perform all supported commands: Write Control Data, Read Status and Scan Status.

4.6.1.1 Write Control Data

A host initiates write operation by asserting the WCTRLDATA signal. When the WCTRLDATA is asserted, that also means that the host data CTLD[15:0], the PHY address FIAD[4:0] and the PHY register address RGAD[4:0] are valid. As soon as the host asserts the WCTRLDATA signal, the MIIM module asserts the BUSY signal to signalize to the host that the write operation is in process. MDOEN is asserted to enable the output line MDO (MDIO) to the PHY. The MIIM module then clocks out the MIIM frame to the PHY on each rising edge of the MDC. The MIIM frame write format conforms to the IEEE 803.2u specification:

- 32-bit long preamble (all ones) if the MIINOPRE bit is not asserted
- 2-bit long Start of frame pattern ST (zero followed by one)
- 2-bit Operation definition (zero-one for write or one-zero for read)

- 5-bit PHY address (FIAD[4:0])
- 5-bit PHY register address RGAD[4:0]
- 2-bit turnaround field TA (one-zero)
- 16-bit data

At the end of the write operation, the BUSY signal is deasserted.

4.6.1.2 Read Status

A host initiates write operation by asserting the RSTAT signal. When RSTAT is asserted, that also means that the PHY address FIAD[4:0] and the PHY register address RGAD[4:0] are valid. As soon as the host asserts the RSTAT signal, the MIIM module asserts the BUSY signal to signalize to the host that the read operation is in process. MDOEN is asserted to enable the output line MDO (MDIO) to the PHY. The MIIM module then clocks out the MIIM frame to the PHY on each rising edge of the MDC and then clocks in the requested data (status). The MIIM read frame format conforms to the IEEE 803.2u specification:

- 32-bit long preamble (all ones) if the MIINOPRE bit is not asserted
- 2-bit long Start of frame pattern ST (zero followed by one)
- 2-bit Operation definition (zero-one for write or one-zero for read)
- 5-bit PHY address (FIAD[4:0])
- 5-bit PHY register address RGAD[4:0]
- 2-bit turnaround field TA (one bit period in which PHY stays in the high-Z state followed by a one-bit period during which the PHY drives a zero on the MDO)
- MIIM deasserts the MDOEN signal that enables the MDI (MDIO works as an input).
- PHY sends the data (status) back to the MIIM module on the data lines PRSD[15:0].

At the end of the read operation, the MIIM deasserts the BUSY signal to indicate to the host that a valid data is on the PRSD[15:0] lines.

4.6.1.3 Scan Status

A host initiates the scan status operation by asserting the SCANSTAT signal. The MIIM performs a continuous read operation of the PHY status register. The PHY is selected by the FIAD[4:0] signals. The link status LinkFail signal is asserted/deasserted by the MIIM module and reflects the link status bit of the PHY status register. The signal NVALID is used for qualifying the validity of the LinkFail signals and the status data PRSD[15:0]. These signals are invalid until the first scan status operation ends.

During the scan status operation, the BUSY signal is asserted until the last read is performed (Scan status operation is stopped).

4.6.2 Operation of the Shift Registers

There are two shift registers in the MII Management Module. Data shift register is used for:

- Shifting out the data to the PHY during the Write Data Control operation
- Shifting in the data during the Read Status operation.
- Shifting out the FIAD[4:0] and the RGAD[4:0] addresses during all operations.

Status shift register contains the data that was latched during the last Read Status Operation. Two additional status signals (LinkFail Status and Status Invalid NVALID) are latched separately from the Status shift register.

When a Scan Operation is requested, the state of the PRSD[15:0] and a MIILF is constantly updated from the selected PHY register. NVALID is used to for qualifying the validity of the PRSD[15:0] and MIILS signals. These signals are invalid until first scan status operation ends.

4.6.3 Operation of the Output Control Module

Output Control Module combines the MDI, MDO and MDOEN signals into a bi-directional MDIO signal that is connected to the external MII PHY. During the Write Control Data operation the MDIO operates as an output from the MIIM module. The signal is used for transferring data from MIIM module to the PHY. During the Read Status operation the MDIO first operates as an output (addressing the PHY and the PHY internal register) and then as an input to the MIIM module (reading the status data). In both cases the most significant bit of the Data is shifted first. When no operation is performed, the MDIO is tri-stated.

4.6.4 Operation of the Clock Generator

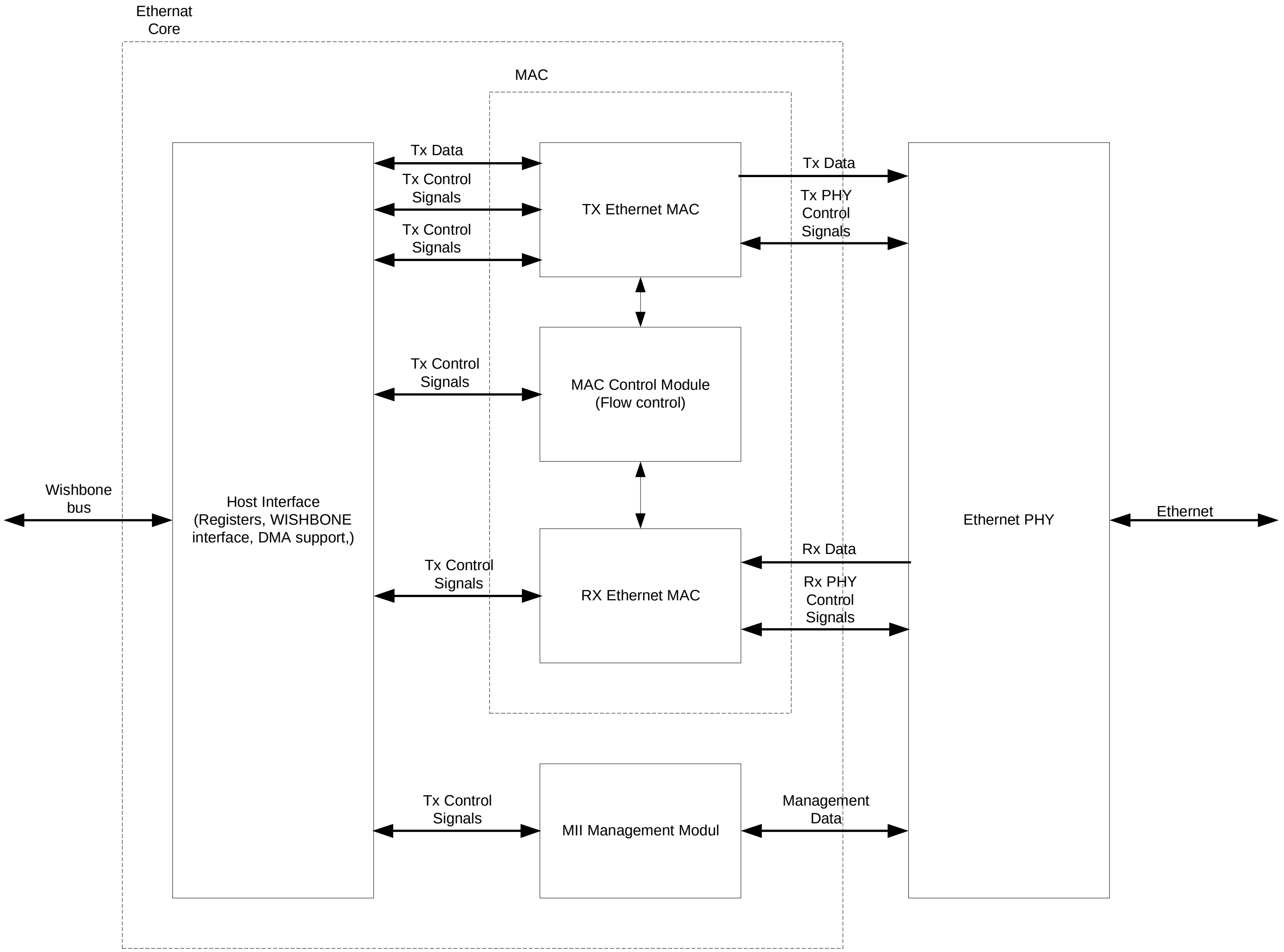
Management Data Clock MDC is a divided host clock. The division factor is set in the MIIMODER register by setting the CLKDIV[7:0] field. (MDC depends on the PHY and can be 2.5 MHz or 12.5 MHz, perhaps something else ☺).

5

Architecture

Ethernet IP Core consists of 5 modules:

- Host Interface and the BD structure
- TX Ethernet MAC (transmit function)
- RX Ethernet MAC (receive function)
- MAC Control Module
- MII Management Module


Ethernat
Core
MAC
Tx Data
Tx Control
Signals
Tx Control
Signals
TX Ethernet MAC
Tx Data
Tx PHY
Control
Signals
Tx Control
Signals
MAC Control Module
(Flow control)
Wishbone
bus
Host Interface
(Registers, WISHBONE
interface, DMA support,)
Ethernet PHY
Ethernet
Rx Data
Tx Control
Signals
RX Ethernet MAC
Rx PHY
Control
Signals
Tx Control
Signals
MII Management Modul
Management
Data

5.1 Host Interface

The host interface is connected to the RISC and the memory through the Wishbone. The RISC writes the data for the configuration registers directly, while the data frames are written to the memory. Frames are accessed through the DMA.

5.2 TX Ethernet MAC

TX Ethernet MAC generates 10BASE-T/100BASE-TX transmit MII nibble data streams in response to the byte streams supplied from the transmit logic (host). It performs the required deferral and back-off algorithms, takes care for the IPG, computes the checksum (FCS) and monitors the physical media (by monitoring Carrier Sense and collision signals).

5.3 RX Ethernet MAC

RX Ethernet MAC interprets 10BASE-T/100BASE-TX MII receive data nibble stream and supplies correctly formed packet byte streams to the host. It searches for the SFD (start frame delimiter) at the beginning of the packet, verifies the FCS and detects any dribble nibbles or receive code violations.

5.4 MAC Control Module

The function of this module is to implement the full-duplex flow control.

The MAC Control Module consists of three sub modules that provide the following functionality:

- Control frame detection
- Control frame generation
- TX/RX Ethernet MAC Interface
- PAUSE Timer
- Slot Timer

5.4.1 Control Frame Detector

Checks the incoming frames for the control frames. Control frames can be discarded or passed to the host. When a PAUSE control frame is detected, it can stop the Tx module from transmitting for a certain period of time.

5.4.2 Control Frame Generator

When there is a need to stop the transmitting station from the transmission (flow control in full duplex mode), a PAUSE control frame can be send to it.

5.4.3 TX/RX Ethernet MAC Interface

MAC Control module is connected between the host interface and the Tx and the Rx MAC modules. Signals from the host are passed by to the Tx MAC in certain occasions and vice versa.

5.4.4 PAUSE Timer

When a PAUSE control frame is received, the pause timer value is written to the PAUSE timer. This prevents the Tx module from transmitting for a »pause timer value« period of slot time.

5.4.5 Slot Timer

Slot timer measures time slots and generate a pulse to the PAUSE timer for every slot time passed by.

5.5 MII Management Module

The function of this module is to control the PHY and to gather the information from it (status).

The MII Management Module consists of four sub modules:

- Operation Control Module
- Output Control Module
- Shift Register
- Clock Generator

5.5.1 Operation Control Module

The function of this module is to perform the following commands:

- Write control data
- Read status
- Scan status

5.5.2 Output Control Module

Controls the signal appearance on the MDO, MCK and MDOEN pins.

5.5.3 Shift Register

Holds the status read from an external PHY.

5.5.4 Clock Generator

Generates an appropriate output clock MCK according to the input host clock and the clock divider bits (CLKDIV[7:0] in the MIIMODER register).