GRETH 10/100 Mbit Ethernet MAC

Based on GRLIB-1.0.9

Jiri Gaisler, Marko Isomaki

1 Introduction

1.1 Overview

The GRETH core implements 10/100 Mbit/s Ethernet Media Access Controller (MAC) with AMBA host interface. The core implements the 802.3-2002 Ethernet standard and supports both MII and RMII PHY interfaces. Receive and transmit data is autonomously transferred between the Ethernet 802.3 Codec and the AMBA AHB bus using DMA transfers. Through the use of receive and transmit descriptors, multiple ethernet packets can be received and transmitted without CPU involvement. The GRETH control registers are accessed through an APB interface. For critical space applications, a fault-tolerant version of GRETH is available with full SEU protection of all RAM blocks.

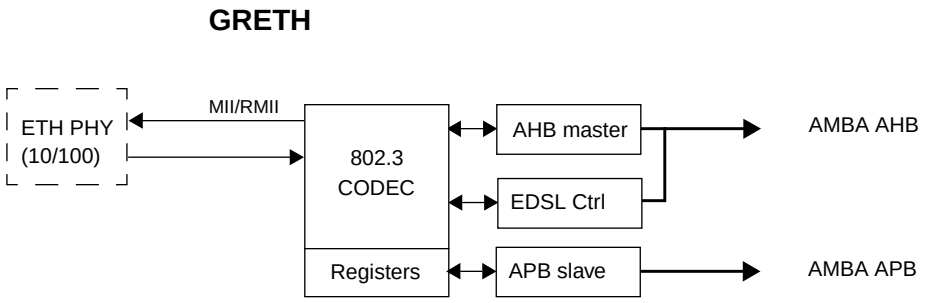


Figure 1. GRETH block diagram

1.2 Implementation characteristics

The GRETH is inherently portable and can be implemented on most FPGA and ASIC technologies. The table below shows the approximate ATOMS/LUT count and frequency for two different GRETH configurations on Altera Stratix, Xilinx Virtex2 and ASIC technologies.

TABLE 1. Implementation characteristics (Cells / RAM blocks / AHB MHz / SPW MHz)

Core configuration	Stratix	Virtex2	ASIC
GRETH	1,800	1,500	10,000 gates
GRSETH + EDCL	3,400	2,800	15,000 gates

The GRETH core is freely available in VHDL source code under the GNU GPL license. It can also be licensed commercially, either stand-alone or as part of the GRLIB IP library.

2 GRETH - Ethernet Media Access Controller (MAC) with EDCL support

2.1 Overview

Gaisler Research's Ethernet Media Access Controller (GRETH) provides an interface between an AMBA-AHB bus and an Ethernet network. It supports 10/100 Mbit speed in both full- and half-duplex. The AMBA interface consists of an APB interface for configuration and control and an AHB master interface which handles the dataflow. The dataflow is handled through DMA channels. There is one DMA engine for the transmitter and one for the receiver. Both share the same AHB master interface. The ethernet interface supports both the MII and RMII interfaces which should be connected to an external PHY. The GRETH also provides access to the MII Management interface which is used to configure the PHY. Optional hardware support for the Ethernet Debug Communication Link (EDCL) protocol is also provided. This is an UDP/IP based protocol used for remote debugging.

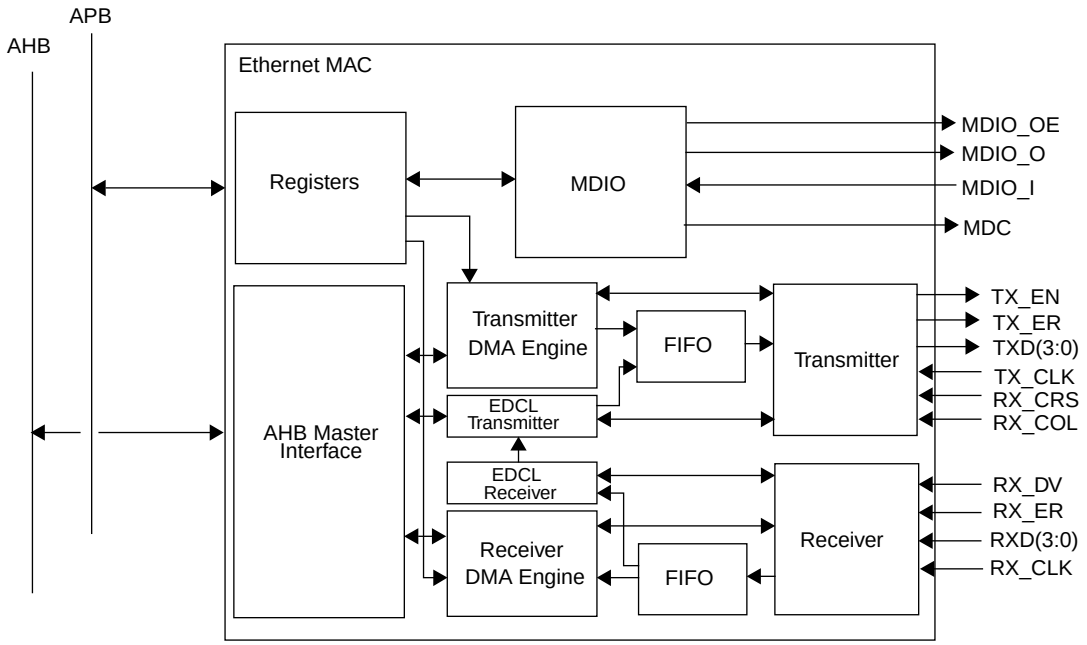


Figure 2. Block diagram of the internal structure of the GRETH.

2.2 Operation

2.2.1 System overview

The GRETH consists 3 functional units: The DMA channels, MDIO interface and the optional Ethernet Debug Communication Link (EDCL).

The main functionality consists of the DMA channels which are used to transfer data between an AHB bus and an Ethernet network. There is one transmitter DMA channel and one Receiver DMA channel. The operation of the DMA channels is controlled through registers accessible through the APB interface.

The MDIO interface is used for accessing configuration and status registers in one or more PHYs connected to the MAC. The operation of this interface is also controlled through the APB interface.

The optional EDCL provides read and write access to an AHB bus through Ethernet. It uses the UDP, IP, ARP protocols together with a custom application layer protocol to accomplish this. The EDCL contains no user accessible registers and always runs in parallel with the DMA channels.

The Media Independent Interface (MII) is used for communicating with the PHY. There is an Ethernet transmitter which sends all data from the AHB domain on the Ethernet using the MII interface. Correspondingly, there is an Ethernet receiver which stores all data from the Ethernet on the AHB bus. Both of these interfaces use FIFOs when transferring the data streams. The GRETH also supports the RMII which uses a subset of the MII signals. More information about the interfaces can be found in section 2.7.

The EDCL and the DMA channels share the Ethernet receiver and transmitter. More information on these functional units is provided in sections 2.3 - 2.6.

2.2.2 Protocol support

The GRETH is implemented according to IEEE standard 802.3-2002. There is no support for the optional control sublayer and no multicast addresses can be assigned to the MAC. This means that packets with type 0x8808 (the only currently defined ctrl packets) are discarded.

2.2.3 Hardware requirements

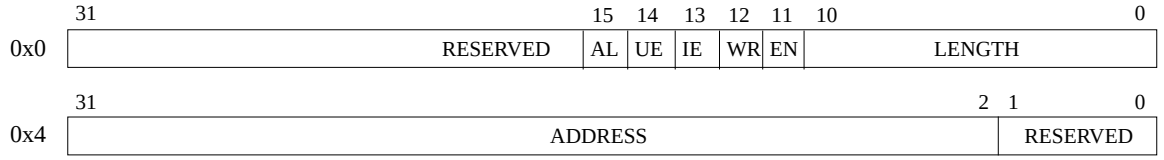
The GRETH is synthesisable with most Synthesis tools. There are three clock domains: The AHB clock, Ethernet Receiver clock and the Ethernet transmitter clock. Both full-duplex and half-duplex operating modes are supported and both can be run in either 10 or 100 Mbit. The system frequency requirement (AHB clock) for 10 Mbit operation is 2.5 MHz and 18 Mhz for 100 Mbit. The 18 Mhz limit was tested on a Xilinx board with a DCM that did not support lower frequencies so it might be possible to run it on lower frequencies. It might also be possible to run the 10 Mbit mode on lower frequencies.

2.3 Tx DMA interface

The transmitter DMA interface is used for transmitting data on an Ethernet network. The transmission is done using descriptors located in memory.

2.3.1 Setting up a descriptor.

A single descriptor is shown in figure 3. The number of bytes to be sent should be set in the length field and the address field should point to the data. The address must be word-aligned. If the interrupt enable (IE) bit is set, an interrupt will be generated when the packet has been sent (this requires that the transmitter interrupt bit in the control register is also set). The interrupt will be generated regardless of whether the packet was transmitted successfully or not. The Wrap (WR) bit is also a control bit that should be set before transmission and it will be explained later in this section.



10 - 0: LENGTH - The number of bytes to be transmitted.

11: Enable (EN) - Set to one to enable the descriptor. Should always be set last of all the descriptor fields.

12: Wrap (WR) - Set to one to make the descriptor pointer wrap to zero after this descriptor has been used. If this bit is not set the pointer will increment by 8. The pointer automatically wraps to zero when the 1 kB boundary of the descriptor table is reached.

13: Interrupt Enable (IE) - Enable Interrupts. An interrupt will be generated when the packet from this descriptor has been sent provided that the transmitter interrupt enable bit in the control register is set. The interrupt is generated regardless if the packet was transmitted successfully or if it terminated with an error.

14: Underrun Error (UE) - The packet was incorrectly transmitted due to a FIFO underrun error.

15: Attempt Limit Error (AL) - The packet was not transmitted because the maximum number of attempts was reached.

31 - 2: Address - Pointer to the buffer area from where the packet data will be loaded.

Figure 3. Transmitter descriptor. Memory offsets are shown in the left margin.

To enable a descriptor the enable (EN) bit should be set and after this is done, the descriptor should not be touched until the enable bit has been cleared by the GRETH.

2.3.2 Starting transmissions

Enabling a descriptor is not enough to start a transmission. A pointer to the memory area holding the descriptors must first be set in the GRETH. This is done in the transmitter descriptor pointer register. The address must be aligned to a 1 kB boundary. Bits 31 to 10 hold the base address of descriptor area while bits 9 to 3 form a pointer to an individual descriptor.The first descriptor should be located at the base address and when it has been used by the GRETH the pointer field is incremented by 8 to point at the next descriptor. The pointer will automatically wrap back to zero when the next 1 kB boundary has been reached (the descriptor at address offset 0x3F8 has been used). The WR bit in the descriptors can be set to make the pointer wrap back to zero before the 1 kB boundary.

The pointer field has also been made writable for maximum flexibility but care should be taken when writing to the descriptor pointer register. It should never be touched when a transmission is active.

The final step to activate the transmission is to set the transmit enable bit in the control register. This tells the GRETH that there are more active descriptors in the descriptor table. This bit should always be set when new descriptors are enabled, even if transmissions are already active. The descriptors must always be enabled before the transmit enable bit is set.

2.3.3 Descriptor handling after transmission

When a transmission of a packet has finished, status is written to the first word in the corresponding descriptor. The Underrun Error bit is set if the FIFO became empty before the packet was completely transmitted while the Alignment Error bit is set if more collisions occurred than allowed. The packet was successfully transmitted only if both of these bits are zero. The other bits in the first descriptor word are set to zero after transmission while the second word is left untouched.

The enable bit should be used as the indicator when a descriptor can be used again, which is when it has been cleared by the GRETH. There are three bits in the GRETH status register that hold transmission status. The Transmitter Error (TE) bit is set each time an transmission ended with an error (when at least one of the two status bits in the transmit descriptor has been set). The Transmitter Interrupt (TI) is set each time a transmission ended successfully.

The transmitter AHB error (TA) bit is set when an AHB error was encountered either when reading a descriptor or when reading packet data. Any active transmissions were aborted and the transmitter was disabled. The transmitter can be activated again by setting the transmit enable register.

2.3.4 Setting up the data for transmission

The data to be transmitted should be placed beginning at the address pointed by the descriptor address field. The GRETH does not add the Ethernet address and type fields so they must also be stored in the data buffer. The 4 B Ethernet CRC is automatically appended at the end of each packet. Each descriptor will be sent as a single Ethernet packet. If the size field in a descriptor is greater than 1514 B, the packet will not be sent.

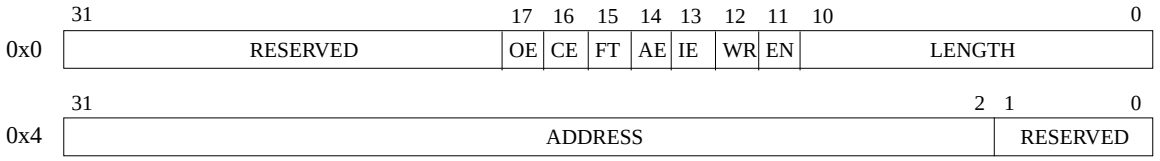
2.4 Rx DMA interface

The receiver DMA interface is used for receiving data from an Ethernet network. The reception is done using descriptors located in memory.

2.4.1 Setting up descriptors

A single descriptor is shown in figure 4. The address field should point to a word-aligned buffer where the received data should be stored. The GRETH will never store more than 1514 B to the buffer. If the interrupt enable (IE) bit is set, an interrupt will be generated when a packet has been

received to this buffer (this requires that the receiver interrupt bit in the control register is also set). The interrupt will be generated regardless of whether the packet was received successfully or not. The Wrap (WR) bit is also a control bit that should be set before the descriptor is enabled and it will be explained later in this section.



10 - 0: LENGTH - The number of bytes received to this descriptor.

11: Enable (EN) - Set to one to enable the descriptor. Should always be set last of all the descriptor fields.

12: Wrap (WR) - Set to one to make the descriptor pointer wrap to zero after this descriptor has been used. If this bit is not set the pointer will increment by 8. The pointer automatically wraps to zero when the 1 kB boundary of the descriptor table is reached.

13: Interrupt Enable (IE) - Enable Interrupts. An interrupt will be generated when a packet has been received to this descriptor provided that the receiver interrupt enable bit in the control register is set. The interrupt is generated regardless if the packet was received successfully or if it terminated with an error.

14: Alignment error (AE) - An odd number of nibbles were received.

15: Frame Too Long (FT) - A frame larger than the maximum size was received. The excessive part was truncated.

16: CRC Error (CE) - A CRC error was detected in this frame.

17: Overrum Error (OE) - The frame was incorrectly received due to a FIFO overrun.

31 - 2: Address - Pointer to the buffer area from where the packet data will be loaded.

Figure 4. Receive descriptor. Memory offsets are shown in the left margin.

2.4.2 Starting reception

Enabling a descriptor is not enough to start reception. A pointer to the memory area holding the descriptors must first be set in the GRETH. This is done in the receiver descriptor pointer register. The address must be aligned to a 1 kB boundary. Bits 31 to 10 hold the base address of descriptor area while bits 9 to 3 form a pointer to an individual descriptor. The first descriptor should be located at the base address and when it has been used by the GRETH the pointer field is incremented by 8 to point at the next descriptor. The pointer will automatically wrap back to zero when the next 1 kB boundary has been reached (the descriptor at address offset 0x3F8 has been used). The WR bit in the descriptors can be set to make the pointer wrap back to zero before the 1 kB boundary.

The pointer field has also been made writable for maximum flexibility but care should be taken when writing to the descriptor pointer register. It should never be touched when reception is active.

The final step to activate reception is to set the receiver enable bit in the control register. This will make the GRETH read the first descriptor and wait for an incoming packet.

2.4.3 Descriptor handling after reception

The GRETH indicates a completed reception by clearing the descriptor enable bit. The other control bits (WR, IE) are also cleared. The number of received bytes is shown in the length field. The parts of the Ethernet frame stored are the destination address, source address, type and data fields. Bits 17-14 in the first descriptor word are status bits indicating different receive errors. All four bits are zero after a reception without errors. The status bits are described in figure 4.

Packets arriving that are smaller than the minimum Ethernet size of 64 B are not considered as a reception and are discarded. The current receive descriptor will be left untouched an used for the first packet arriving with an accepted size. The TS bit in the status register is set each time this event occurs.

If a packet is received with an address not accepted by the MAC, the IA status register bit will be set.

Packets larger than maximum size cause the FT bit in the receive descriptor to be set. The length field is not guaranteed to hold the correct value of received bytes. The counting stops after the word containing the last byte up to the maximum size limit has been written to memory.

The address word of the descriptor is never touched by the GRETH.

2.4.4 Reception with AHB errors

If an AHB error occurs during a descriptor read or data store, the Receiver AHB Error (RA) bit in the status register will be set and the receiver is disabled. The current reception is aborted. The receiver can be enabled again by setting the Receive Enable bit in the control register.

2.5 MDIO Interface

The MDIO interface provides access to PHY configuration and status registers through a two-wire interface which is included in the MII interface. The GRETH provided full support for the MDIO interface. If it is not needed in a design it can be removed with a generic (see section 2.8).

The MDIO interface can be used to access from 1 to 32 PHY containing 1 to 32 16-bit registers. A read transfer i set up by writing the PHY and register addresses to the MDIO Control register and setting the read bit. This caused the Busy bit to be set and the operation is finished when the Busy bit is cleared. If the operation was successful the Linkfail bit is zero and the data field contains the read data. An unsuccessful operation is indicated by the Linkfail bit being set. The data field is undefined in this case.

A write operation is started by writing the 16-bit data, PHY address and register address to the MDIO Control register and setting the write bit. The operation is finished when the busy bit is cleared and it was successful if the Linkfail bit is zero.

2.6 Ethernet Debug Communication Link (EDCL)

The EDCL provides access to an on-chip AHB bus through Ethernet. It uses the UDP, IP and ARP protocols together with a custom application layer protocol. The application layer protocol uses an ARQ algorithm to provide reliable AHB instruction transfers. Through this link, a read or write transfer can be generated to any address on the AHB bus. The EDCL is optional and must be enabled with a generic.

2.6.1 Operation

The EDCL receives packets in parallel with the MAC receive DMA channel. It uses a separate MAC address which is used for distinguishing EDCL packets from packets destined to the MAC DMA channel. The EDCL also has an IP address which is set through generics. Since ARP packets use the Ethernet broadcast address, the IP-address must be used in this case to distinguish between EDCL ARP packets and those that should go to the DMA-channel. Packets that are determined to be EDCL packets are not processed by the receive DMA channel.

When the packets are checked to be correct, the AHB operation is performed. The operation is performed with the same AHB master interface that the DMA-engines use. The replies are automatically sent by the EDCL transmitter when the operation is finished. It shares the Ethernet transmitter with the transmitter DMA-engine but has higher priority.

2.6.2 EDCL protocols

The EDCL accepts Ethernet frames containing IP or ARP data. ARP is handled according to the protocol specification with no exceptions.

IP packets carry the actual AHB commands. The EDCL expects an Ethernet frame containing IP, UDP and the EDCL specific application layer parts. Table 2 shows the IP packet required by the EDCL. The contents of the different protocol headers can be found in TCP/IP literature.

TABLE 2. The IP packet expected by the EDCL.

Ethernet Header	IP Header	UDP Header	2 B Offset	4 B Control word	4 B Address	Data 0 - 242 4B Words	Ethernet CRC

The following is required for successful communication with the EDCL: A correct destination MAC address as set by the generics, an Ethernet type field containing 0x0806 (ARP) or 0x0800 (IP). The IP-address is then compared with the value determined by the generics for a match. The IP-header checksum and identification fields are not checked. There are a few restrictions on the IP-header fields. The version must be four and the header size must be 5 B (no options). The protocol field must always be 0x11 indicating a UDP packet. The length and checksum are the only IP fields changed for the reply.

The EDCL only provides one service at the moment and it is therefore not required to check the UDP port number. The reply will have the original source port number in both the source and destination fields. UDP checksum are not used and the checksum field is set to zero in the replies.

The UDP data field contains the EDCL application protocol fields. Table 3 shows the application protocol fields (data field excluded) in packets received by the EDCL. The 16-bit offset is used to align the rest of the application layer data to word boundaries in memory and can thus be set to any value. The R/W field determines whether a read (0) or a write(1) should be per-

TABLE 3. The EDCL application layer fields in received frames.

16-bit Offset	14-bit Sequence number	1-bit R/W	10-bit Length	7-bit Unused

formed. The length field contains the number of bytes to be read or written. If R/W is one the data field shown in Table 2 contains the data to be written. If R/W is zero the data field is empty in the received packets. Table 4 shows the application layer fields of the replies from the EDCL. The length field is always zero for replies to write requests. For read requests it contains the number of bytes of data contained in the data field.

TABLE 4. The EDCL application layer fields in transmitted frames.

16-bit Offset	14-bit sequence number	1-bit ACK/NAK	10-bit Length	7-bit Unused

The EDCL implements a Go-Back-N algorithm providing reliable transfers. The 14-bit sequence number in received packets are checked against an internal counter for a match. If they do not match, no operation is performed and the ACK/NAK field is set to 1 in the reply frame. The reply frame contains the internal counter value in the sequence number field. If the sequence number matches, the operation is performed, the internal counter is incremented, the internal counter value is stored in the sequence number field and the ACK/NAK field is set to 0 in the reply. The length field is always set to 0 for ACK/NAK=1 frames. The unused field is not checked and is copied to the reply. It can thus be set to hold for example some extra id bits if needed.

2.7 Media Independent Interfaces

There are several interfaces defined between the MAC sublayer and the Physical layer. The GRETH supports two of them: The Media Independent Interface (MII) and the Reduced Media Independent Interface (RMII).

The MII was defined in the 802.3 standard and is most commonly supported. The ethernet interface have been implemented according to this specification. It uses 16 signals.

The RMII was developed to meet the need for an interface allowing Ethernet controllers with smaller pin counts. It uses 6 (7) signals which are a subset of the MII signals. Table 5 shows the mapping betweem the RMII signals and the GRLIB MII interface.

TABLE 5. Signal mappings between RMII and the GRLIB MII interface.

RMII	**MII**
txd[1:0]	txd[1:0]
tx_en	tx_en
crs_dv	rx_crs
rxd[1:0]	rxd[1:0]
ref_clk	rmii_clk
rx_er	not used

2.8 Configuration options

The GRETH has the following configuration options (VHDL generics):

TABLE 6. GRETH configuration options (VHDL generics)

Generic	Function	Allowed range	Default
hindex	AHB master index.	0 - NAHBMST-1	0
pindex	APB slave index	0 - NAPBSLV-1	0
paddr	Addr field of the APB bar.	0 - 16#FFF#	0
pmask	Mask field of the APB bar.	0 - 16#FFF#	16#FFF#
pirq	Interrupt line used by the GRETH.	0 - NAHBIRQ-1	0
memtech	Memory technology used for the FIFOs.	0 - NTECH	inferred
ifg_gap	Number of ethernet clock cycles used for one interframe gap. Default value as required by the standard. Do not change unless you know what your doing.	1 - 255	24
attempt_limit	Maximum number of transmission attempts for one packet. Default value as required by the standard.	1 - 255	16
backoff_limit	Limit on the backoff size of the backoff time. Default value as required by the standard. Sets the number of bits used for the random value. Do not change unless you know what your doing.	1 - 10	10
slot_time	Number of ethernet clock cycles used for one slot-time. Default value as required by the ethernet standard. Do not change unless you know what you are doing.	1 - 255	128
mdcscaler	Sets the divisor value use to generate the mdio clock (mdc). The mdc frequency will be clk/ (2*(mdcscaler+1)).	0 - 255	25
enable_mdio	Enable the Management interface,	0 - 1	0
fifosize	Sets the size in 32-bit words of the receiver and transmitter FIFOs.	4 - 32	8
nsync	Number of synchronization registers used.	1 - 2	2
edcl	Enable EDCL.	0 - 1	0
edclbufsize	Select the size of the EDCL buffer in kB.	1 - 64	1
macaddrh	Sets the upper 24 bits of the EDCL MAC address.*)	0 - 16#FFFFFF#	16#00005E#
macaddrl	Sets the lower 24 bits of the EDCL MAC address. *)	0 - 16#FFFFFF#	16#000000#
ipaddrh	Sets the upper 16 bits of the EDCL IP address reset value.	0 - 16#FFFF#	16#C0A8#
ipaddrl	Sets the lower 16 bits of the EDCL IP address reset value.	0 - 16#FFFF#	16#0035#
phyrstadr	Sets the reset value of the PHY address field in the MDIO register.	0 - 31	0
rmii	Selects the desired PHY interface. 0 = MII, 1 = RMII.	0 - 1	0

*) Not all addresses are allowed and most NICs and protocol implementations will discard frames with illegal addresses silently. Consult network literature if unsure about the addresses.

2.9 Vendor and device id

The module has vendor id 0x01 (Gaisler Research) and device id 0x1D. For description of vendor and device ids see GRLIB IP Library User's Manual.

2.10 Registers

The GRETH is programmed through 7 registers mapped into APB address space.

TABLE 7. GRETH registers

Register	APB Address offset
Control register	0x0
Status/Interrupt-source register	0x4
MAC Address MSB	0x8
MAC Address LSB	0xC
MDIO Control/Status	0x10
Transmit descriptor pointer	0x14
Receiver descriptor pointer	0x18
EDCL IP	0x1C

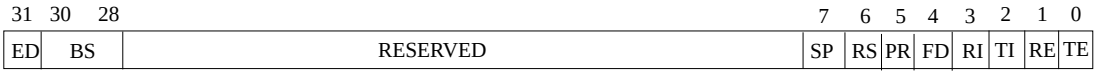


Figure 5. GRETH control register.

0: Transmit Enable (TE) - Should be written with a one each time new descriptors are enabled. As long as this bit is one the GRETH will read new descriptors and as soon as it encounters a disabled descriptor it will stop until TE is set again. This bit should be written with a one after the new descriptors have been enabled. Reset value: '0'.

1: Receive Enable (RE) - Should be written with a one each time new descriptors are enabled. As long as this bit is one the GRETH will read new descriptors and as soon as it encounters a disabled descriptor it will stop until TE is set again. This bit should be written with a one after the new descriptors have been enabled. Reset value: '0'.

2: Transmitter Interrupt (TI) - Enable Transmitter Interrupts. An interrupt will be generated each time a packet is transmitted when this bit is set. The interrupt is generated regardless if the packet was transmitted successfully or if it terminated with an error. Not Reset.

3: Receiver Interrupt (RI) - Enable Receiver Interrupts. An interrupt will be generated each time a packet is received when this bit is set. The interrupt is generated regardless if the packet was received successfully or if it terminated with an error. Not Reset.

4: Full Duplex (FD) - If set, the GRETH operates in full-duplex mode otherwise it operates in half-duplex. Not Reset.

5: Promiscuous Mode (PM) - If set, the GRETH operates in promiscuous mode which means it will receive all packets regardless of the destination address. Not Reset.

6: Reset (RS) - A one written to this bit resets the GRETH core. Self clearing.

7: Speed (SP) - Sets the current speed mode. 0 = 10 Mbit, 1 = 100 Mbit. Only used in RMII mode (rmii = 1). A default value is automatically read from the PHY after reset.

30 - 28: EDCL Buffer Size (BS) - Shows the amount of memory used for EDCL buffers. 0 = 1 kB, 1 = 2 kB, ..., 6 = 64 kB.

31: EDCL Available (ED) - Set to one if the EDCL is available.

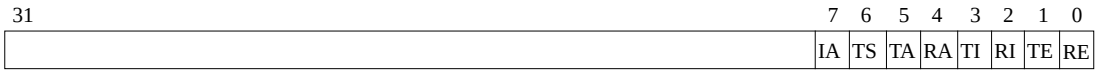


Figure 6. GRETH status register

0: Receiver Error (RE) - A packet has been received which terminated with an error. Cleared when written with a one. Not Reset.

1: Transmitter Error (TE) - A packet was transmitted which terminated with an error. Cleared when written with a one. Not Reset.

2: Receiver Interrupt (RI) - A packet was received without errors. Cleared when written with a one. Not Reset.

3: Transmitter Interrupt (TI) - A packet was transmitted without errors. Cleared when written with a one. Not Reset.

4: Receiver AHB Error (RA) - An AHB error was encountered in receiver DMA engine. Cleared when written with a one. Not Reset.

5: Transmitter AHB Error (TA) - An AHB error was encountered in transmitter DMA engine. Cleared when written with a one. Not Reset.

6: Too Small (TS) - A packet smaller than the minimum size was received. Cleared when written with a one. Reset value: ‘0’.

7: Invalid Address (IA) - A packet with an address not accepted by the MAC was received. Cleared when written with a one. Reset value: ‘0’.

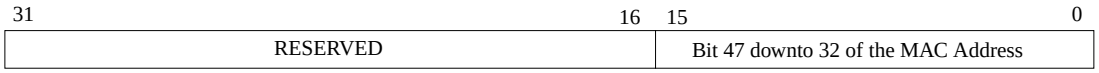


Figure 7. MAC Address MSB.

31 - 16: The two most significant bytes of the MAC Address. Not Reset.

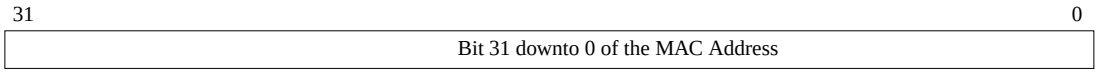


Figure 8. MAC Address LSB.

31 - 0: The 4 least significant bytes of the MAC Address. Not Reset.

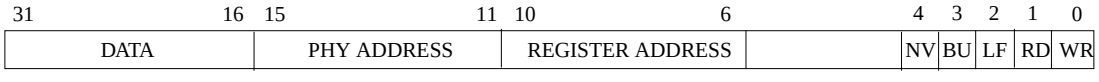


Figure 9. GRETH MDIO ctrl/status register.

0: Write (WR) - Start a write operation on the management interface. Data is taken from the Data field. Reset value: ‘0’.

1: Read (RD) - Start a read operation on the management interface. Data is stored in the data field. Reset value: ‘0’.

2: Linkfail (LF) - When an operation completes (BUSY = 0) this bit is set if a functional management link was not detected. Not Reset.

3: Busy (BU) - When an operation is performed this bit is set to one. As soon as the operation is finished and the management link is idle this bit is cleared. Reset value: ‘0’.

4: Not valid (NV) - When an operation is finished (BUSY = 0) this bit indicates whether valid data has been received that is, the data field contains correct data. Not Reset.

10 - 6: Register Address - This field contains the address of the register that should be accessed during a write or read operation. Not Reset.

15 - 11: PHY Address - This field contains the address of the PHY that should be accessed during a write or read operation. Not Reset.

31 - 16: Data - Contains data read during a read operation and data that is transmitted is taken from this field. Not Reset.

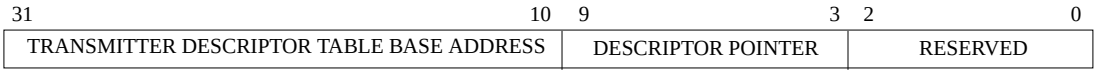


Figure 10. GRETH transmitter descriptor table base address register.

31 - 10: Base address to the transmitter descriptor table.Not Reset.

9 - 3: Pointer to individual descriptors. Automatically incremented by the Ethernet MAC.

2 - 0: Reserved. Reads as zeroes.

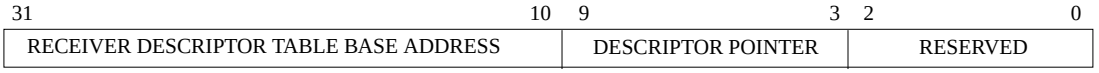


Figure 11. GRETH receiver descriptor table base address register.

31 - 10: Base address to the receiver descriptor table.Not Reset.

9 - 3: Pointer to individual descriptors. Automatically incremented by the Ethernet MAC.

2 - 0: Reserved. Reads as zeroes.

Figure 12. GRETH EDCL IP register.

31 - 0: EDCL IP address. Reset value is set with the ipaddrh and ipaddrl generics.

2.11 Software drivers

Drivers for the GRETH MAC is provided for the following operating systems: RTEMS, eCos, uClinux and Linux-2.6. The drivers are freely available in full source code under the GPL license from Gaisler Research's web site (http://gaisler.com/).

2.12 Signal description

GRETH signals are described in table 8.

TABLE 8. GRETH signals.

Signal name	Field	Type	Function	Active
RST	N/A	Input	Reset	Low
CLK	N/A	Input	Clock	-
AHBMI	*	Input	AMB master input signals	-
AHBMO	*	Output	AHB master output signals	-
APBI	*	Input	APB slave input signals	-
APBO	*	Output	APB slave output signals	-
ETHI	*	Input	Ethernet MII input signals.	-
ETHO	*	Output	Ethernet MII output signals.	-

* see GRLIB IP Library User's Manual

2.13 Library dependencies

Table 9 shows libraries that should be used when instantiating a GRETH.

TABLE 9. Library dependencies

Library	Package	Imported unit(s)	Description
GRLIB	AMBA	Signals	AMBA signal definitions
GAISLER	ETHERNET_MAC	Signals, component	GRETH component declarations, GRETH signals.
GAISLER	NET	Signals	Ethernet signals.

2.14 GRETH instantiation.

This examples shows how a GRETH can be instantiated.

```
library ieee;
use ieee.std_logic_1164.all;

library grlib;
use grlib.amba.all;
use grlib.tech.all;
library gaisler;
use gaisler.ethernet_mac.all;

entity greth_ex is
  port (
    clk  : in std_ulogic;
    rstn : in std_ulogic;

    -- ethernet signals
    ethi :: in  eth_in_type;
    etho :  in  eth_out_type
    );
end;
```

```
architecture rtl of greth_ex is

  -- AMBA signals
  signal apbi  : apb_slv_in_type;
  signal apbo  : apb_slv_out_vector := (others => apb_none);
  signal ahbmi : ahb_mst_in_type;
  signal ahbmo : ahb_mst_out_vector := (others => ahbm_none);

begin

  -- AMBA Components are instantiated here
  ...

  -- GRETH
  e1 : greth
    generic map(
     hindex       => 0,
     pindex       => 12,
     paddr        => 12,
     pirq         => 12,
     memtech      => inferred,
     mdcscaler    => 50,
     enable_mdio  => 1,
     fifosize     => 32,
     nsync        => 1,
     edcl         => 1,
     edclbufsz    => 8,
     macaddrh     => 16#00005E#,
     macaddrl     => 16#00005D#,
     ipaddrh      => 16#c0a8#,
     ipaddrl      => 16#0035#)
 port map(
   rst          => rstn,
   clk          => clk,
   ahbmi        => ahbmi,
   ahbmo        => ahbmo(0),
   apbi         => apbi,
   apbo         => apbo(12),
   ethi         => ethi,
   etho         => etho
   );
end;
```