Please note that Cypress is an Infineon Technologies Company.
The document following this cover page is marked as “Cypress” document as this is the company that originally developed the product. Please note that Infineon will continue to offer the product to new and existing customers as part of the Infineon product portfolio.

Continuity of document content
The fact that Infineon offers the following product as part of the Infineon product portfolio does not lead to any changes to this document. Future revisions will occur when appropriate, and any changes will be set out on the document history page.

Continuity of ordering part numbers
Infineon continues to support existing part numbers. Please continue to use the ordering part numbers listed in the datasheet for ordering.



32-Bit Microcontroller FM4 Family Peripheral Manual Ethernet Part

Doc. No. 002-04963 Rev.*C

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709
www.cypress.com

Copyrights

© Cypress Semiconductor Corporation, 2014-2018. This document is the property of Cypress Semiconductor Corporation and its subsidiaries, including Spansion LLC (“Cypress”). This document, including any software or firmware included or referenced in this document (“Software”), is owned by Cypress under the intellectual property laws and treaties of the United States and other countries worldwide. Cypress reserves all rights under such laws and treaties and does not, except as specifically stated in this paragraph, grant any license under its patents, copyrights, trademarks, or other intellectual property rights. If the Software is not accompanied by a license agreement and you do not otherwise have a written agreement with Cypress governing the use of the Software, then Cypress hereby grants you a personal, non-exclusive, nontransferable license (without the right to sublicense) (1) under its copyright rights in the Software (a) for Software provided in source code form, to modify and reproduce the Software solely for use with Cypress hardware products, only internally within your organization, and (b) to distribute the Software in binary code form externally to end users (either directly or indirectly through resellers and distributors), solely for use on Cypress hardware product units, and (2) under those claims of Cypress’s patents that are infringed by the Software (as provided by Cypress, unmodified) to make, use, distribute, and import the Software solely for use with Cypress hardware products. Any other use, reproduction, modification, translation, or compilation of the Software is prohibited.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS DOCUMENT OR ANY SOFTWARE OR ACCOMPANYING HARDWARE, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. No computing device can be absolutely secure. Therefore, despite security measures implemented in Cypress hardware or software products, Cypress does not assume any liability arising out of any security breach, such as unauthorized access to or use of a Cypress product. In addition, the products described in these materials may contain design defects or errors known as errata which may cause the product to deviate from published specifications. To the extent permitted by applicable law, Cypress reserves the right to make changes to this document without further notice. Cypress does not assume any liability arising out of the application or use of any product or circuit described in this document. Any information provided in this document, including any sample design information or programming code, is provided only for reference purposes. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. Cypress products are not designed, intended, or authorized for use as critical components in systems designed or intended for the operation of weapons, weapons systems, nuclear installations, life-support devices or systems, other medical devices or systems (including resuscitation equipment and surgical implants), pollution control or hazardous substances management, or other uses where the failure of the device or system could cause personal injury, death, or property damage (“Unintended Uses”). A critical component is any component of a device or system whose failure to perform can be reasonably expected to cause the failure of the device or system, or to affect its safety or effectiveness. Cypress is not liable, in whole or in part, and you shall and hereby do release Cypress from any claim, damage, or other liability arising from or related to all Unintended Uses of Cypress products. You shall indemnify and hold Cypress harmless from and against all claims, costs, damages, and other liabilities, including claims for personal injury or death, arising from or related to any Unintended Uses of Cypress products.

Cypress, the Cypress logo, Spansion, the Spansion logo, and combinations thereof, WICED, PSoC, CapSense, EZ-USB, F-RAM, and Traveo are trademarks or registered trademarks of Cypress in the United States and other countries. For a more complete list of Cypress trademarks, visit cypress.com. Other names and brands may be claimed as property of their respective owners.

Arm and Cortex are registered trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere.

Preface

Thank you for your continued use of Cypress products.

Read this manual and Data Sheet thoroughly before using products in this family.

Purpose of This Manual and Intended Readers

This manual explains the functions and operations of this family and describes how it is used. The manual is intended for engineers engaged in the actual development of products using this family.

* This manual explains the configuration and operation of the peripheral functions, but does not cover the specifics of each device in the family.
  Users should refer to the respective data sheets of devices for device-specific details.

* Whether a peripheral function is on board or not is dependent on product type. See data sheets for details.

Sample Programs and Development Environment

Cypress offers sample programs free of charge for using the peripheral functions of the FM3 family. Cypress also makes available descriptions of the development environment required for this family. Feel free to use them to verify the operational specifications and usage of this Cypress microcontroller.

Microcontroller support information

https://community.cypress.com/community/MCU

Note:

Note that the sample programs are subject to change without notice. Since they are offered as a way to demonstrate standard operations and usage, evaluate them sufficiently before running them on your system.
Cypress assumes no responsibility for any damage that may occur as a result of using a sample program.

Related Manuals

The manuals related to this family are listed below. See the manual appropriate to the applicable conditions.

The contents of these manuals are subject to change without notice. Contact us to check the latest versions available.

Peripheral Manual

FM4 Family Peripheral Manual (002-04856)
  Called "Peripheral Manual" hereafter
FM4 Family Peripheral Manual Timer Part (002-04858)
  Called "Timer Part" hereafter
FM4 Family Peripheral Manual Analog Macro Part (002-04860)
  Called "Analog Macro Part" hereafter
FM4 Family Peripheral Manual Communication Macro Part (002-04862)
  Called "Communication Macro Part" hereafter
FM4 Family Peripheral Manual Ethernet Part (this manual)
  Called "Ethernet Part" hereafter

Data Sheet

For details about device-specific, electrical characteristics, package dimensions, ordering information etc., see the following document.

32-bit Microcontroller FM4 Family Data Sheet

Note:
The data sheets for each series are provided.
See the appropriate data sheet for the series that you are using.

CPU Programming Manual

For details about Arm Cortex-M4F core, see the following documents that can be obtained from

http://www.arm.com/.

Cortex-M4 Technical Reference Manual
Arm v7-M Architecture Application Level Reference Manual

Flash Programming Manual

For details about the functions and operations of the built-in flash memory, see the following document.

FM4 Family Flash Programming Manual

Note:
The flash programming manual for each series are provided.
See the appropriate flash programming manual for the series that you are using.

How to Use This Manual

Finding a Function

The following methods can be used to search for the explanation of a desired function in this manual:

Search from the table of the contents
The table of the contents lists the manual contents in the order of description.

About the Chapters

Basically, this manual explains Ethernet Part.

Terminology

This manual uses the following terminology.

Term	Explanation
Word	Indicates access in units of 32 bits.
Half word	Indicates access in units of 16 bits.
Byte	Indicates access in units of 8 bits.

Notations

■ The notations in bit configuration of the register explanation of this manual are written as follows.

- bit: bit number
- Field: bit field name
- Attribute: Attributes for read and write of each bit
  - R: Read only
  - W: Write only
  - R/W: Readable/Writable
  - -: Undefined
- Initial value: Initial value of the register after reset
  - 0: Initial value is 0
  - 1: Initial value is 1
  - X: Initial value is undefined

■ The multiple bits are written as follows in this manual.

- Example: bit7:0 indicates the bits from bit7 to bit0

■ The values such as for addresses are written as follows in this manual.

- Hexadecimal number: "0x" is attached in the beginning of a value as a prefix (example: 0xFFFF)
- Binary number: "0b" is attached in the beginning of a value as a prefix (example: 0b1111)
- Decimal number: Written using numbers only (example: 1000)

The Target Products in This Manual

Table 1 TYPE3-M4 Product List

Description in this manual*	Flash memory size		
	2 Mbytes	1.5 Mbytes	1 Mbytes
TYPE3-M4	S6E2CCAL S6E2CCAJ S6E2CCAH	S6E2CC9L S6E2CC9J S6E2CC9H	S6E2CC8L S6E2CC8J S6E2CC8H
	S6E2C2AL S6E2C2AJ S6E2C2AH	S6E2C29L S6E2C29J S6E2C29H	S6E2C28L S6E2C28J S6E2C28H

*It categorizes each product in FM4 peripheral manual.

Table 2 TYPE5-M4 Product List

Description in this manual*	Flash memory size	
	1 Mbytes	512Kbytes
TYPE5-M4	S6E2GM8J S6E2GM8H	S6E2GM6J S6E2GM6H
	S6E2GK8J S6E2GK8H	S6E2GK6J S6E2GK6H
	S6E2G28J S6E2G28H	S6E2G26J S6E2G26H

*It categorizes each product in FM4 peripheral manual.

Contents

CHAPTER 1: Ethernet 11
1. Overview 12
1.1. Overview 12
1.2. Block Diagram 12
1.3. Description of Each Block 13
2. I/O Signal of Ethernet 14
2.1. Microcontroller External Pin 14
2.2. Microcontroller Internal Connection Pin 18
3. Ethernet-MAC Setup Control Procedure 19
4. Ethernet System Control Block Register 20
4.1. Mode Select Register (ETH_MODE) 21
4.2. Clock Gating Register (ETH_CLKG) 23
CHAPTER 2: Ethernet-MAC 25
1. General Description 26
2. Block Configuration 27
3. Architecture 28
3.1. Pin Functions 29
3.2. AHB Application Host Interface 31
3.3. DMA Controller 32
3.3.1. Descriptor Controller 32
3.3.2. Initialization of DMA Controller 33
3.3.3. Transmission 35
3.3.4. Reception 40
3.3.5. Interrupts 43
3.3.6. Error Response to DMA 43
3.3.7. CRC 43
3.4. Checksum Engine 44
3.4.1. Transmit Checksum Offroad Engine 44
3.4.2. Receive Checksum Offroad Engine 46
3.5. Energy Efficient Ethernet 47
3.6. MAC Management Counters 50
3.7. Station Management Agent 52
3.8. IEEE1588 54
4. Registers 62
4.1. GMAC Register 0 (MCR) 68
4.2. GMAC Register 1 (MFFR) 71
4.3. GMAC Register 2, 3 (MHTRH, MHTRL) 74
4.4. GMAC Register 4 (GAR) 75
4.5. GMAC Register 5 (GDR) 77
4.6. GMAC Register 6 (FCR) 78
4.7. GMAC Register 7 (VTR) 80
4.8. GMAC Register 10 (RWFFR) 81
4.9. GMAC Register 11 (PMTR) 83
4.10. GMAC Register 12 (LPICSR) 85
4.11. GMAC Register 13 (LPITCR) 87

4.12. GMAC Register 14 (ISR) 88
4.13. GMAC Register 15 (IMR) 90
4.14. GMAC Register 16 (MAR0H) 91
4.15. GMAC Register 17 (MAR0L) 92
4.16. GMAC Register 18,20,22,..., 542 (MAR1H,2H,3H,...,31H) 93
4.17. GMAC Register 19,21,23,25,..., 543 (MAR1L,2L,3L,...,31L) 95
4.18. GMAC Register 54 (RGSR) 96
4.19. GMAC Register 448 (TSCR) 97
4.20. GMAC Register 449 (SSIR) 100
4.21. GMAC Register 450 (STSR) 101
4.22. GMAC Register 451 (STNR) 102
4.23. GMAC Register 452 (STSUR) 103
4.24. GMAC Register 453 (STNUR) 104
4.25. GMAC Register 454 (TSAR) 105
4.26. GMAC Register 455 (TTSR) 106
4.27. GMAC Register 456 (TTNR) 107
4.28. GMAC Register 457 (STHWSR) 108
4.29. GMAC Register 458 (TSR) 109
4.30. GMAC Register 459 (PPSCR) 110
4.31. GMAC Register 460 (ATNR) 112
4.32. GMAC Register 461 (ATSR) 113
4.33. DMA Register 0 (BMR) 114
4.34. DMA Register 1 (TPDR) 117
4.35. DMA Register 2 (RPDR) 118
4.36. DMA Register 3 (RDLAR) 119
4.37. DMA Register 4 (TDLAR) 120
4.38. DMA Register 5 (SR) 121
4.39. DMA Register 6 (OMR) 125
4.40. DMA Register 7 (IER) 128
4.41. DMA Register 8 (MFBOCR) 131
4.42. DMA Register 9 (RIWTR) 132
4.43. DMA Register 11 (AHBSR) 133
4.44. DMA Register 18 (CHTDR) 134
4.45. DMA Register 19 (CHRDR) 135
4.46. DMA Register 20 (CHTBAR) 136
4.47. DMA Register 21 (CHRBAR) 137
4.48. MMC Register List 138
4.49. GMAC Register.64 (mmc_cntl) 142
4.50. GMAC Register.65 (mmc_intr_rx) 143
4.51. GMAC Register.66 (mmc_intr_tx) 145
4.52. GMAC Register.67 (mmc_intr_mask_rx) 147
4.53. GMAC Register.68 (mmc_intr_mask_tx) 149
4.54. GMAC Register.128 (mmc_ipc_intr_mask_rx) 151
4.55. GMAC Register.130 (mmc_ipc_intr_rx) 153

5. Descriptors 155
5.1. Transmit Enhanced Descriptor 156
5.1.1. Transmit Enhanced Descriptor 0 (TDES0) 157
5.1.2. Transmit Enhanced Descriptor 1 (TDES1) 160
5.1.3. Transmit Enhanced Descriptor 2 (TDES2) 161
5.1.4. Transmit Enhanced Descriptor 3 (TDES3) 162
5.1.5. Transmit Enhanced Descriptor 6 (TDES6) 163
5.1.6. Transmit Enhanced Descriptor 7 (TDES7) 164
5.2. Receive Enhanced Descriptor 165
5.2.1. Receive Enhanced Descriptor 0 (RDES0) 166
5.2.2. Receive Enhanced Descriptor 1 (RDES1) 168
5.2.3. Receive Enhanced Descriptor 2 (RDES2) 169
5.2.4. Receive Enhanced Descriptor 3 (RDES3) 170
5.2.5. Receive Enhanced Descriptor 4 (RDES4) 171
5.2.6. Receive Enhanced Descriptor 6 (RDES6) 173
5.2.7. Receive Enhanced Descriptor 7 (RDES7) 174
6. Programming Guide 175
6.1. DMA Initialization/ Descriptor 175
6.2. GMAC Initialization 176
6.3. Normal Receive and Transmit Operation 176
6.4. Stop and Start Operation 177
6.5. Link down-up sequence 177
6.6. Programming Guideline for IEEE Time Stamping 177
6.7. Programming Guideline for Energy Efficient Ethernet 178
6.8. Transmittion to Standby Mode and SYS_CLK Stop 179
Appendixes 180
1. Major Changes 180
Revision History 181
Document Revision History 181

CHAPTER 1: Ethernet

This chapter describes the configuration of Ethernet related blocks.

1. Overview
2. I/O Signal of Ethernet
3. Ethernet-MAC Setup Control Procedure
4. Ethernet System Control Block Register

1. Overview

This section describes the overview of Ethernet function and Ethernet-MAC setup control procedure.

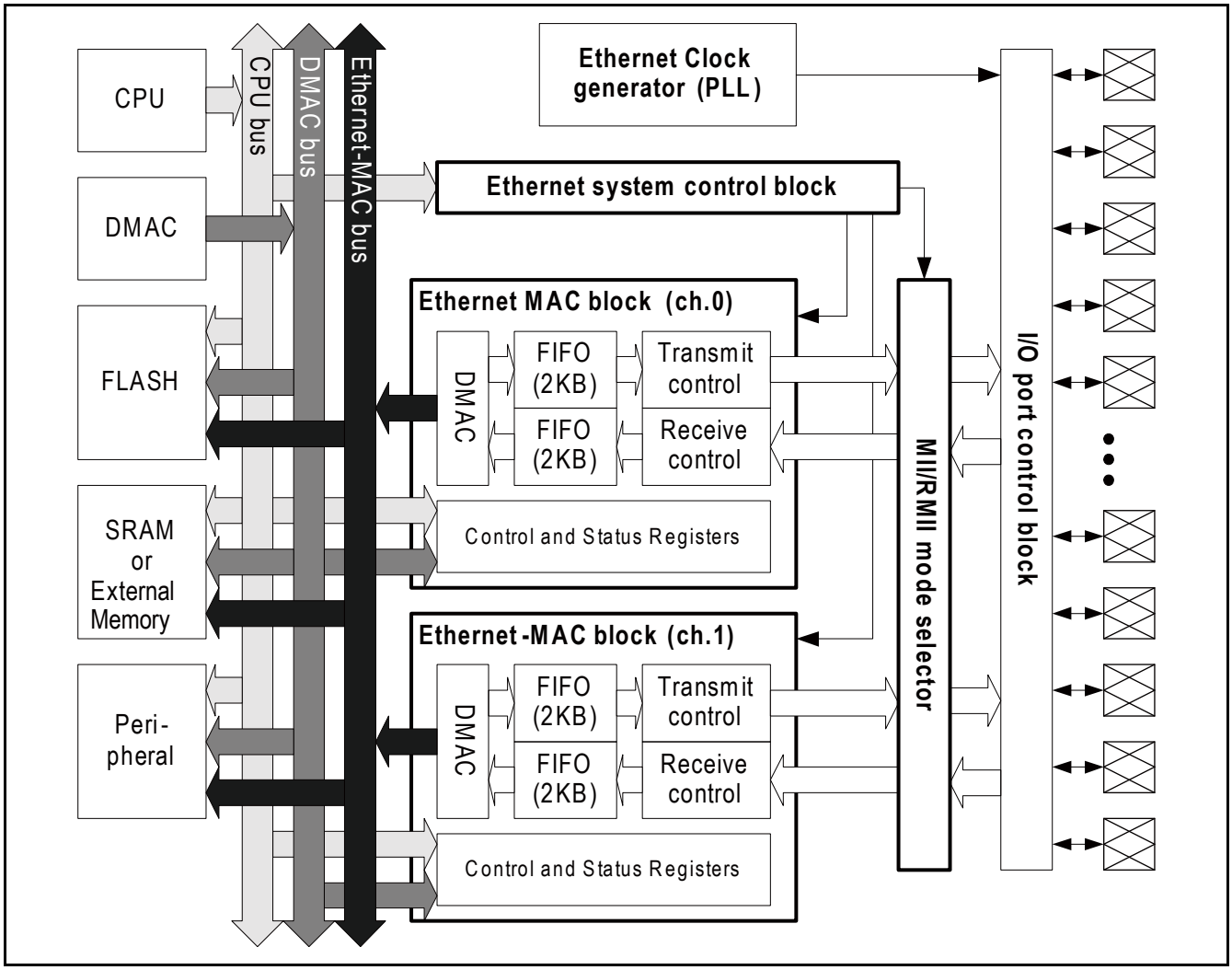
1.1 Overview

The Ethernet function of this family is composed of Ethernet-MAC block and its peripheral circuit. This chapter describes Ethernet-MAC the peripheral circuit and Ethernet-MAC setup control procedure.

1.2 Block Diagram

Figure 1-1 shows the configuration diagram of the Ethernet-MAC and peripheral circuit.

Figure 1-1 Configuration Diagram of Ethernet-MAC and Peripheral Circuit



1.3 Description of Each Block

Ethernet-MAC Block

Ethernet-MAC is the block that controls Ethernet communication protocol control. Although there are two Ethernet-MACs (ch.0 and ch.1) in the block diagram, some products have only ch.0 and other products have both ch.0 and ch.1. The products that have only one Ethernet-MAC cannot use the ch.1 side. Those that have two Ethernet-MACs can use both of them separately.

Each Ethernet-MAC is composed of send control block, receive control block, FIFO memory, and dedicated DMAC. It controls the operation by instructing the control register block (Control and Status Registers) from the CPU. The dedicated DMAC uses a descriptor on the memory that is created by the CPU and executes send and receive data transfers. For details on Ethernet-MAC, see the chapter Ethernet-MAC.

To operate each Ethernet-MAC, execute the following Ethernet-MAC setup by register writing to the Ethernet system control block beforehand.

1. Start supplying clocks for each Ethernet-MAC
2. Select MII/RMII mode
3. Issue a hardware reset to each Ethernet-MAC and restart the Ethernet-MAC

For the information above, see 3 Ethernet-MAC Setup Control Procedure.

Each Ethernet-MAC executes an initial setting to instruct the start of the operation after releasing the hardware reset of the above setup. Internal registers of each Ethernet-MAC have spots that select MII/RMII. These are set separately from the above setup procedure.

Ethernet system control block

The Ethernet system control block controls the peripheral circuit of Ethernet-MAC and the setup of each Ethernet-MAC. The setup control procedure is described in 3 Ethernet-MAC Setup Control Procedure. Also, the register functions of the Ethernet system control block are described in 4 Ethernet System Control Block Register.

If each Ethernet-MAC is not operated, you can execute low power consumption of the microcontroller by stopping the supply of clock signals. If it is switched to stop, the supply of clock signals setup control becomes necessary.

MII/RMII Mode Selector

MII/RMII mode selector is the selector block that connects the external PHY interface signal of Ethernet-MAC with the I/O port control block. Based on the register setting of the Ethernet system control block, the interface mode (MII or RMII) to use is selected.

I/O Port Control Block

Pin functions of the external pins of the microcontroller are used with the peripheral function block pins other than Ethernet (e.g. GPIO). The setting of control registers in the I/O port block is required to use the external pins of the microcontroller as external PHY interface pins of the Ethernet.
When the microcontroller is in low power consumption mode (Stop mode or Timer mode), I/O signals of PHY interface pin must be enabled (called the last retention state) to receive WakeUP packet from external PHY.
For details on the I/O port control block, see the chapter "I/O PORT" in the peripheral manual.

Ethernet Clock Generator

The clock signals generated by the Ethernet clock generator can be output to a microcontroller pin. If the above clock output signals are generated by the PLL, input clock oscillation accuracy of general external PHY devices cannot be guaranteed. So, please be careful. For details on the Ethernet clock generator, see the chapter USB/Ethernet Clock Generation Block in the Communication Macro Part".

2. I/O Signal of Ethernet

This section describes the connection of I/O signals related to the Ethernet function.

2.1 Microcontroller External Pin

External PHY Interface Signal

The functions of the Microcontroller external pin are also used by Ethernet-MAC ch.0 and ch.1 and the PHY interface signal pin of MII/RMII. Which interface signal is used is decided at the setup of the Ethernet system control block.

Table 2-1 shows the correspondence of each microcontroller external pin name of both the product with one Ethernet-MAC and the products with two Ethernet-MACs and PHY interface signals of each Ethernet-MAC.

Table 2-1 Correspondence of Microcontroller External Pin Name and PHY Interface Signals

Microcontroller External Pin Name		MII (using Ch.0)	RMII (using Ch.0)	RMII (using Ch.0 and Ch.1)	Remarks
Product with Only Ch.0	Product with Both Ch.0 and Ch.1				
E_RXCK_REFCK	E_RXCK0_REFCK	ch.0 RX_CLK	ch.0 REF_CLK	ch.0 REF_CLK ch.1 REF_CLK	*1
E_RX00	E_RX00	ch.0 RXD [0]	ch.0 RXD [0]	ch.0 RXD [0]	
E_RX01	E_RX01	ch.0 RXD [1]	ch.0 RXD [1]	ch.0 RXD [1]	
E_RX02	E_RX02_RX10	ch.0 RXD [2]	Do not use	ch.1 RXD [0]	
E_RX03	E_RX03_RX11	ch.0 RXD [3]	Do not use	ch.1 RXD [1]	
E_RXDV	E_RXDV0	ch.0 RX_DV	ch.0 CRS_DV	ch.0 CRS_DV	
E_RXER	E_RXER0_RXDV1	ch.0 RX_ER	Do not use	ch.1 CRS_DV	*2
E_TCK	E_TCK0_MDC1	ch.0 TX_CLK	Do not use	ch.1 MDC	
E_TX00	E_TX00	ch.0 TXD [0]	ch.0 TXD [0]	ch.0 TXD [0]	
E_TX01	E_TX01	ch.0 TXD [1]	ch.0 TXD [1]	ch.0 TXD [1]	
E_TX02	E_TX02_TX10	ch.0 TXD [2]	Do not use	ch.1 TXD [0]	
E_TX03	E_TX03_TX11	ch.0 TXD [3]	Do not use	ch.1 TXD [1]	
E_TXEN	E_TXEN0	ch.0 TX_EN	ch.0 TX_EN	ch.0 TX_EN	
E_TXER	E_TXER0_TXEN1	ch.0 TX_ER	Do not use	ch.1 TX_EN	*3
E_CRS	E_CRS0	ch.0 CRS	Do not use	Do not use	
E_COL	E_COL0	ch.0 COL	Do not use	Do not use	
E_MDC	E_MDC0	ch.0 MDC	ch.0 MDC	ch.0 MDC	
E_MDIO	E_MDIO0	ch.0 MDIO	ch.0 MDIO	ch.0 MDIO	
-	E_MDIO1	Do not use	Do not use	ch.1 MDIO	

*1: This pin is used as the RX_CLK input in the case of MII. In the case of RMII, it is used as REF_CLK input. If both ch.0 and ch.1 of the Ethernet-MAC are used, it is common to both channels.

*2: In the case of RMII, it is not necessary to connect it because the RX_ER input from external PHY is not used.

*3: In the case of MII, TX_ER output executes High level output is only in the LPI mode of EEE (Energy Efficient Ethernet). If EEE is not used, connection to the PHY is not necessary.

If both ch.0 and ch.1 of Ethernet-MAC are used, MII cannot be selected.

Example of Connection to External PHY

The following figures the connection to external PHY devices.

Although not indicated in the Figure 2-1 to Figure 2-3, it recommends connecting pull-up resistance to the MDIO signal.

Using Ethernet-MAC (ch.0) and connecting to PHY in MII mode (Figure 2-1)
Using Ethernet-MAC (ch.0) and connecting to PHY in RMII mode (Figure 2-2)
Using Ethernet-MAC (ch.0 and ch.1) and connecting to two PHYs in RMII mode (Figure 2-3)

Figure 2-1 MII Mode Connection Diagram (using Ethernet-MAC Ch.0)

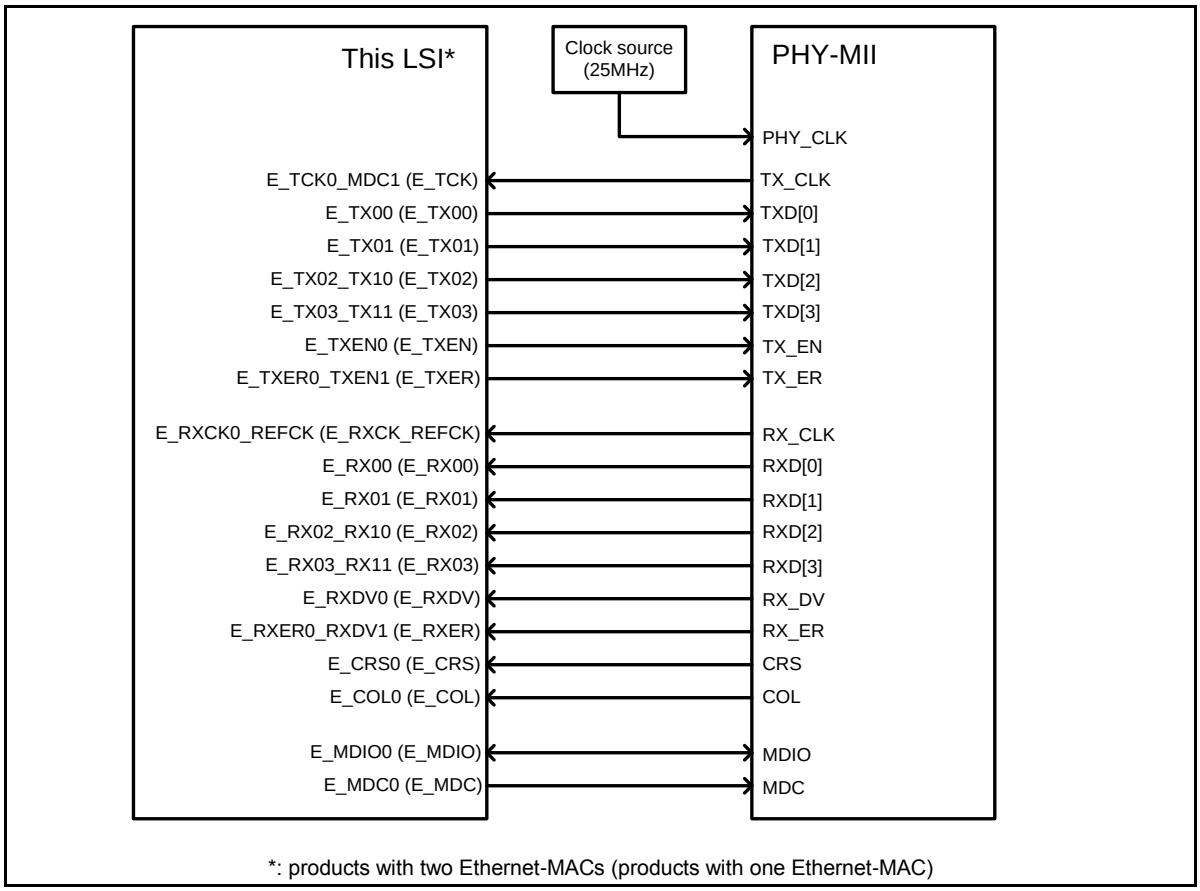


Figure 2-2 RMII Mode Connection Diagram (using Ethernet-MAC Ch.0)

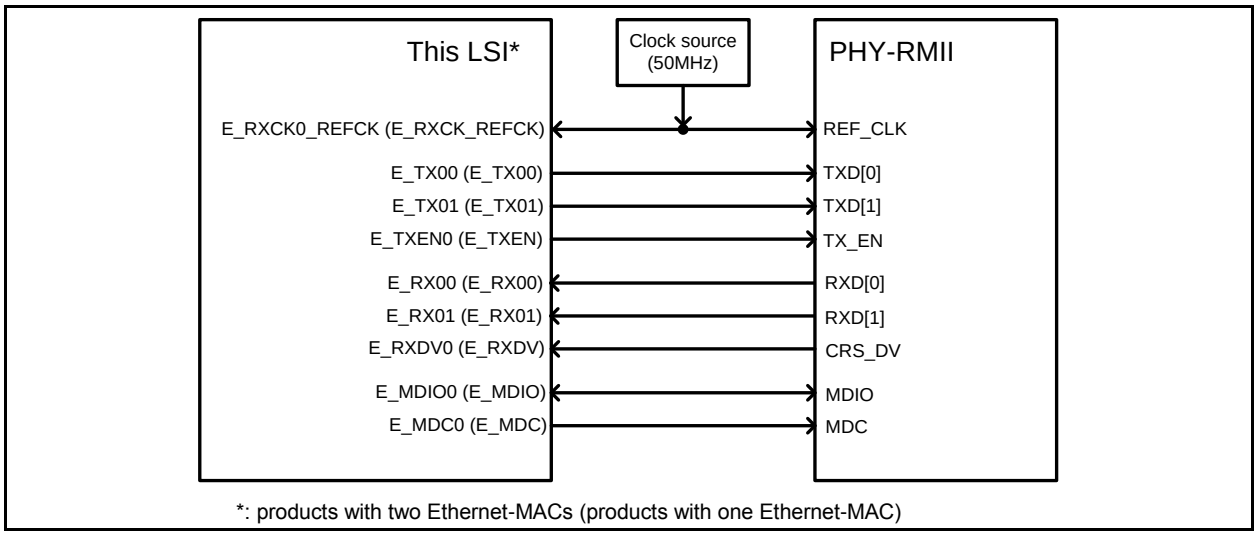
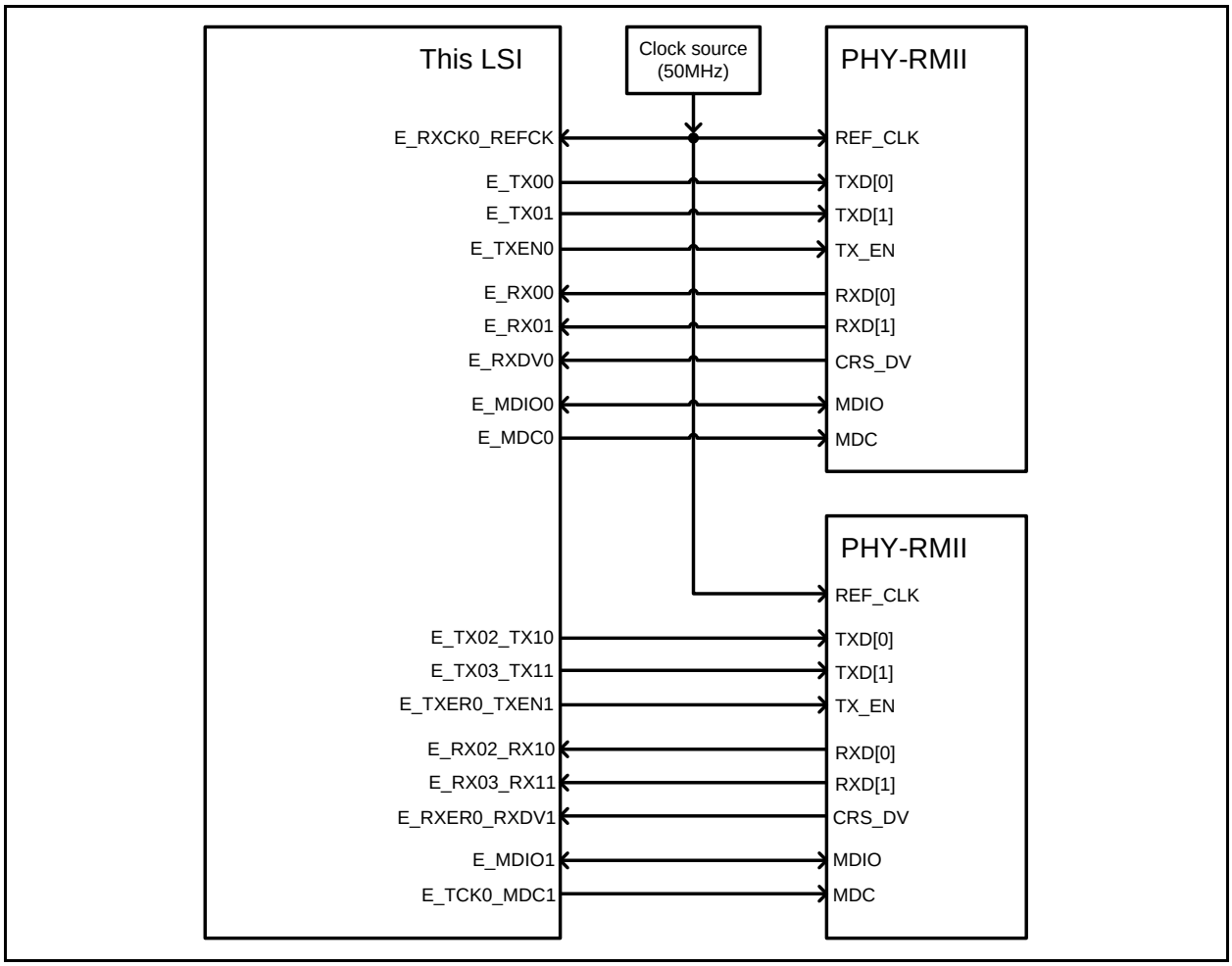


Figure 2-3 RMII Mode Connection Diagram (using Ethernet-MAC Ch.0 and Ch.1)



Other Microcontroller External Pins

Table 2-2 shows the microcontroller external pins that are other than those shown in Table 2-1 and related to Ethernet function.

Table 2-2 Other Microcontroller External Pins

Microcontroller External Pin Name	Function
E_PPS0_PPS1 *	A pulse in each one-second of system time counter of the PTP function is output from the Ethernet-MAC block. Selection of either ch.0 output or ch.1 output can be determined by the ETH_MODE.PTPSEL register.
E_COUT	A pin to output clock signals is generated by the Ethernet clock generator.

*: Products with only one Ethernet-MAC use the E_PPS pin. Ch.1 output cannot be selected.

2.2 Microcontroller Internal Connection Pin

System Clock Signal and AHB Bus Interface

Each Ethernet-MAC is connected to the CPU and memory via AHB bus. When using each Ethernet-MAC, the AHB clock (HCLK) must be set to more than 25 MHz.

The AHB clock (HCLK) is connected to the source clock (PTP_CLK) of thesystem time counter of each Ethernet-MAC.

Interrupt Signal Interface

Interrupt signals output from each Ethernet-MAC block are shown in Table 2-3. These interrupts are connected to NVIC. For these interrupts, see the chapter Interrupts in the Peripheral Manual.

Table 2-3 Interrupt Signal from Ethernet-MAC

Interrupt Signal Name	Function
MAC0_INT_SBD	The interrupt asserted along with the end of DMAC processing of Ethernet-MAC ch.0.
MAC0_INT_PMT	The interrupt asserted when valid WAKE-UP frame reception of Ethernet-MAC ch.0 is detected.
MAC0_INT_LPI	The interrupt asserted when Ethernet-MAC ch.0 ends LPI state.
MAC1_INT_SBD	The interrupt asserted along with the end of DMAC processing of Ethernet-MAC ch.1.
MAC1_INT_PMT	The interrupt asserted when valid WAKE-UP frame reception of Ethernet-MAC ch.1 is detected.

Ethernet-MAC ch.1 generates no LPI interrupts because it does not have the MII mode.

3. Ethernet-MAC Setup Control Procedure

This section describes the Ethernet-MAC setup control procedure.

About the Setup

After selecting the I/O port block, execute the following procedure to set up each Ethernet-MAC. It includes clock supply to each Ethernet-MAC, selection of MII/RMII mode, and issue/release of hardware reset to each Ethernet-MAC. After the setup is complete, execute the initial setting for each Ethernet-MAC to start operation.

Procedure when MII Mode is Selected

1. Write ETH_CLKG.MACEN0 = 1 and ETH_CLKG.MACEN1 = 0. Start clock supply to Ethernet-MAC (ch.0).
2. Write ETH_MODE.IFMODE = 0, ETH_MODE.RST0 = 1, and ETH_MODE.RST1 = 0. Select MII and issue a hardware reset to Ethernet-MAC (ch.0).
3. At this point, clock signals (RX_CLK and TX_CLK) must be input from the external PHY. If the clock signals have not been input, wait until they are input.
4. Write ETH_MODE.IFMODE = 0, ETH_MODE.RST0 = 0, and ETH_MODE.RST1 = 0. Select MII and release a hardware reset of Ethernet-MAC (ch.0).

Procedure when RMII Mode (using only ch.0) is Selected

1. Write ETH_CLKG.MACEN0 = 1, ETH_CLKG.MACEN1 = 0. Start clock supply to Ethernet-MAC (ch.0).
2. Write ETH_MODE.IFMODE = 1, ETH_MODE.RST0 = 1, and ETH_MODE.RST1 = 0. Select RMII and issue a hardware reset to Ethernet-MAC (ch.0).
3. At this point, clock signal (REF_CLK) must be input from external PHY. If the clock signal has not been input, wait until it is input.
4. Write ETH_MODE.IFMODE = 1, ETH_MODE.RST0 = 0, and ETH_MODE.RST1 = 0. Select RMII and release a hardware reset of Ethernet-MAC (ch.0).

Procedure when RMII Mode (using Both ch.0 and ch.1) is Selected

1. Write ETH_CLKG.MACEN0 = 1 and ETH_CLKG.MACEN1 = 1. Start clock supply to Ethernet-MAC (ch.0 and ch.1).
2. Write ETH_MODE.IFMODE = 1, ETH_MODE.RST0 = 1, and ETH_MODE.RST1 = 1. Select RMII and issue a hardware reset to Ethernet-MAC (ch.0 and ch.1).
3. At this point, clock signal (REF_CLK) must be input from external PHY. If the clock signal has not been input, wait until it is input.
4. Write ETH_MODE.IFMODE = 1, ETH_MODE.RST0 = 0, and ETH_MODE.RST1 = 0. Select RMII and release a hardware reset of Ethernet-MAC (ch.0 and ch.1).

*: ETH_MODE.PTPSEL can be selected arbitrarily according to the channel of the PTP signal being output.

4. Ethernet System Control Block Register

This section describes Ethernet system control block registers.

Table 4-1 shows the list of Ethernet system control block registers.

Table 4-1 Ethernet System Control Block Register List

Offset Address	Register Abbreviation	Register Name	Reference
0x00	ETH_MODE	Mode select register	4.1
0x08	ETH_CLKG	Clock gating register	4.2

Ethernet system control block base address: 0x40066000

4.1 Mode Select Register (ETH_MODE)

ETH_MODE controls the external interface mode of the Ethernet.

Register configuration

bit	31	30	29	28	27	26	25	24
Field	Reserved			PPSSEL	Reserved			
Attribute	R	R	R	R/W	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

bit	23	22	21	20	19	18	17	16
Field	Reserved							
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

bit	15	14	13	12	11	10	9	8
Field	Reserved						RST1	RST0
Attribute	R	R	R	R	R	R	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	Reserved							IFMODE
Attribute	R	R	R	R	R	R	R	R/W
Initial value	0	0	0	0	0	0	0	0

Register function

[bit31:29] Reserved

Write 0 when writing. 0 is read when reading.

[bit28] PPSSEL

Select either of the system time counter pulse outputs of the Ethernet-MAC PTP function to output to E_PPS0_PPS1 pin.

bit	Operation
Write 0	Select Ethernet-MAC (ch.0).
Write 1	Select Ethernet-MAC (ch.1).
Read	Read the register setting value.

[bit27:10] Reserved

Write 0 when writing. 0 is read when reading.

[bit9] RST1

Control hardware reset signal against Ethernet-MAC (ch.1).

bit	Operation
Write 0	Release a hardware reset against Ethernet-MAC (ch.1).
Write 1	Issue a hardware reset against Ethernet-MAC (ch.1).
Read	Read the register setting value.

[bit8] RST0

Control the hardware reset signal against Ethernet-MAC (ch.0).

bit	Operation
Write 0	Release a hardware reset against Ethernet-MAC (ch.0).
Write 1	Issue a hardware reset against Ethernet-MAC (ch.0).
Read	Read the register setting value.

[bit7:1] Reserved

Write 0 when writing. 0 is read when reading.

[bit0] IFMODE

Select the connection of the MII/RMII mode selector.

bit	Operation
Write 0	Select MII.
Write 1	Select RMII.
Read	Read the register setting value.

4.2 Clock Gating Register (ETH_CLKG)

ETH_CLKG controls the clock supply to each Ethernet-MAC.

Register configuration

bit	31	30	29	28	27	26	25	24
Field	Reserved							
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

bit	23	22	21	20	19	18	17	16
Field	Reserved							
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

bit	15	14	13	12	11	10	9	8
Field	Reserved							
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	Reserved						MACEN	
Attribute	R	R	R	R	R	R	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

Register function

[bit31:2] Reserved

Write 0 when writing. 0 is read when reading.

[bit1:0] MACEN

Select the system clock supply to Ethernet-MAC.

bit1:0	Products that have one Ethernet-MAC	Products that have two Ethernet-MACs
Write 00	Stop supplying clocks to Ethernet-MAC (ch.0)	Stop supplying clocks to Ethernet-MAC (ch.0 and ch.1)
Write 01	Start supplying clocks to Ethernet-MAC (ch.0)	Start supplying clocks to Ethernet-MAC (ch.0) Stop supplying clocks to Ethernet-MAC (ch.1)
Write 11	Setting is prohibited	Start supplying clocks to Ethernet-MAC (ch.0 and ch.1)
Write 10	Setting is prohibited	Setting is prohibited.
Read	Read the register setting value.	Read the register setting value.

CHAPTER 2: Ethernet-MAC

This document provides specifications of Ethernet-MAC.

1. General Description
2. Block Configuration
3. Architecture
4. Registers
5. Descriptors
6. Programming Guide

1. General Description

Ethernet-MAC consists of four blocks shown below. This section explains the main features of each block.

GMAC: Ethernet Media Access Controller

- IEEE802.3-2005 compliant
- Supports data transfer speeds of 10/100 Mbps
- IEEE802.3 compliant MII interface
- Supports full-duplex and half-duplex operation at 10/100 Mbps
- Supports the CSMA/CD protocol with half-duplex operation
- Supports the back pressure in half-duplex mode
- Supports IEEE 802.3x flow control with full-duplex operation
- Automatically generates CRC and data padding
- Supports Jumbo frame
- Supports various flexible address filtering modes
- Supports the internal loopback for debugging on the MII
- Supports IEEE802.3Q VLAN packet
- Supports Wake-On-LAN (Remote Wake-Up, Magic Packet)
- Supports Checksum Offload
- Supports IEEE 802.3-az-2010 for Energy Efficient Ethernet (EEE)
- Supports RMII interface
- IEEE1588-2008 compliant
- Supports MAC Management Counter

MTL: MAC Transaction Layer

- Supports 2K-bytes of Transmit FIFO and 2K-bytes Receive FIFO
- Handles automatic retransmission of Collision frames for transmission
- Discards frames on late collision and Underrun conditions

DMA: DMA Controller

- Descriptor-system DMA
- Supports little-endian for transmit and receive data paths

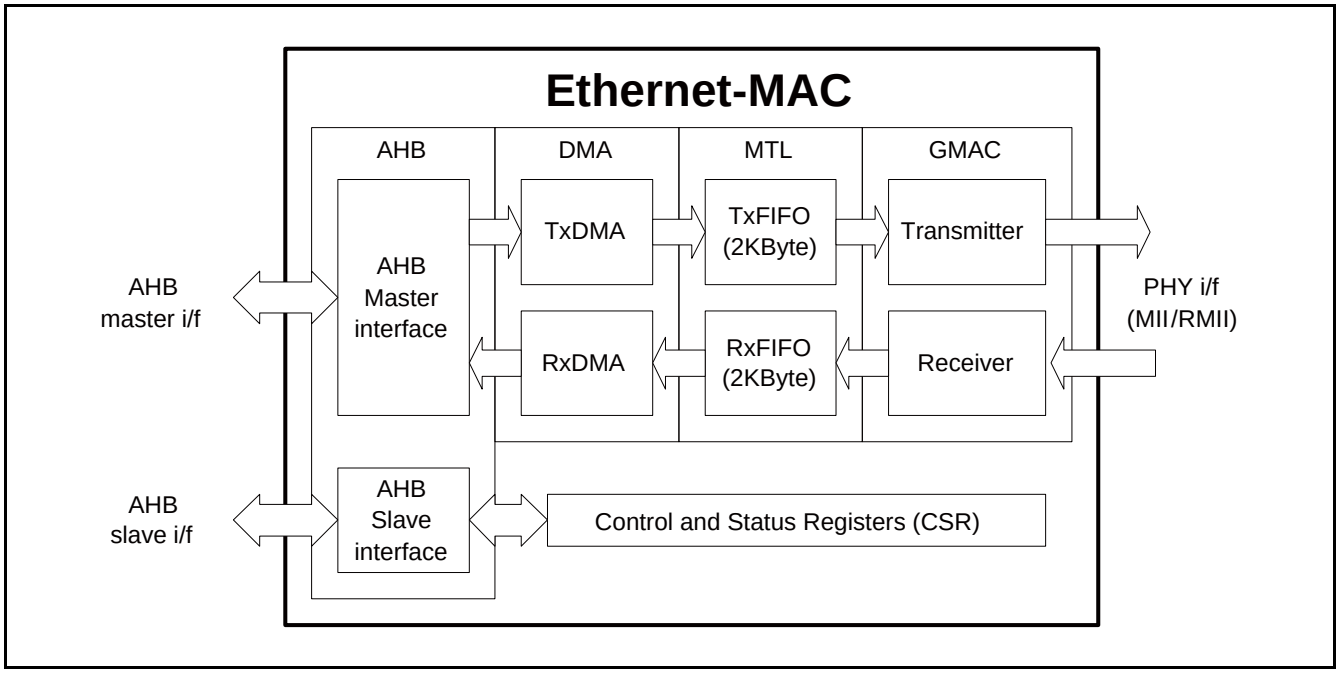
AHB: AHB Interface

- AMBA Specification Rev.2.0 compliant
- Address and data of 32-bit width
- Supports multi-layer bus

2. Block Configuration

Figure 2-1 shows the block diagram of the Ethernet-MAC.

Figure 2-1 Block Diagram of Ethernet-MAC



This section explains each function block.

AHB: AHB Interface

This is a function block controlling the interface with system bus (AHB). Internal DMA has the AHB master interface and can directly access to the host memory via the system bus. The AHB slave interface performs as an interface with the host CPU and Control Status registers (CSR).

DMA: DMA Controller

Transmit DMA (TxDMA) and Receive DMA (RxDMA) are configured to operate independently from each other. According to the contents of the descriptor table that is built to host memory, the DMA processes the data transfer.

MTL: MAC Transaction Layer

2Kbyte FIFO memory is equipped respectively for buffering of Transmit data/Receive data. MTL is a function block controlling the FIFO, and processes automatic re-transmission, discarding of frames, etc.

GMAC: Ethernet Media Access Controller

This is a function block realizing the communication protocol of the Ethernet. The PHY interface supports MII/RMII.

3. Architecture

This section explains the architecture of the Ethernet-MAC.

3.1. Pin Functions

3.2. AHB Application Host Interface

3.3. DMA Controller

3.4. Checksum Engine

3.5. Energy Efficient Ethernet

3.6. MAC Management Counters

3.7. Station Management Agent

3.8. IEEE 1588

3.1 Pin Functions

This section explains the pin functions of the Ethernet-MAC.

PHY Interface Pins

Table 3-1 shows the PHY interface pins. GMAC supports both MII and RMII interfaces. Pins indicated with the ○ mark of the column MII/RMII in the table are required to be connected to the PHY.

Table 3-1 PHY Interface Pins

Signal Pin Name	I/O	Functions	MII	RMII	Remarks
RX_CLK	IN	Receive clock	○	×	For 100 Mbps: 25 MHz For 10 Mbps: 2.5 MHz
RXD[0]	IN	Receive data 0	○	○	
RXD[1]	IN	Receive data 1	○	○	
RXD[2]	IN	Receive data 2	○	×	
RXD[3]	IN	Receive data 3	○	×	
RX_DV	IN	Receive data valid	○	×	
RX_ER	IN	Receive error detection	○	×	Connection not required for RMII
TX_CLK	IN	Transmit clock	○	×	For 100 Mbps: 25 MHz For 10 Mbps: 2.5 MHz
TXD[0]	OUT	Transmit data 0	○	○	
TXD[1]	OUT	Transmit data 1	○	○	
TXD[2]	OUT	Transmit data 2	○	×	
TXD[3]	OUT	Transmit data 3	○	×	
TX_EN	OUT	Transmit data valid	○	○	
TX_ER	OUT	Transmit error	○	×	Energy Efficient Ethernet Connect only with compatible PHY
CRS	IN	Carrier detection	○	×	
COL	IN	Collision detection	○	×	
MDC	OUT	Management clock	○	○	Generates by dividing the SYS_CLK input.
MDIO	IO	Management data	○	○	This is controlled bi-directionally by external pins. Pull-up is recommended.
REF_CLK	IN	Reference clock	×	○	50 MHz clock input
CRS_DV	IN	Carrier detection/data valid	×	○	

Host Interfaces and Other Pins

Table 3-2 shows host interfaces and other pins. Pins are limited as needed to explain the operations and functions of this circuit.

Table 3-2 Host Interfaces and Other Pins

Pin Name	I/O	Function	Description
SYS_CLK	IN	System clock	Clock (HCLK) on the AHB bus is connected. This is the reference clock for the transfer operation of DMA. The MDC of the management interface is generated by dividing SYS_CLK. Always supply the HCLK of 25 MHz or higher.
PTP_CLK	IN	PTP clock	This is the reference clock input of the PTP system counter module. The AHB clock (HCLK) on the microcontroller is connected.
INT_SBD	OUT	Ethernet-MAC interrupt	This is the interrupt signal to notify the various events from Ethernet-MAC. This is connected to NVIC on the microcontroller.
INT_PMT	OUT	WakeUp interrupt	This is the interrupt signal to notify the event that the GMAC receiver receives a WakeUp packet from PHY. This allows the microcontroller to recover from standby mode.
INT_LPI	OUT	LPI interrupt	This is the interrupt signal to notify the event that the GMAC receiver receives a notification of recovery from LPI (Low Power Idle) state from PHY. This allows the microcontroller to recover from standby mode.
PTPPPS	OUT	Second count output	This is the second count output of the PTP system time counter module. This can be output to external pins of the microcontroller. For details, see "4.30. GMAC Register 459 (PPSCR)".

3.2 AHB Application Host Interface

This section explains the AHB interface of the Ethernet-MAC.
The AHB Master interface controls data transfers by converting the internal DMA request cycles into AHB cycles.
The AHB slave interface provides to access to GMAC form the host CPU and the CSR (Control and Status Register) space of the DMA.
The AHB Master Interface can control data transfers while the AHB Slave interface accesses CSR space.

AHB Master Interface

- You can choose fixed burst length (SINGLE, INCR4, INCR8, INCR16) or unspecified burst length (SINGLE, INCR) transfers or a mix of fixed and unspecified burst transfers by programming the FB or MB bits in the DMA Bus Mode register.
- When fixed burst length is chosen, the AHB master always initiates a burst with SINGLE, INCR4, INCR8 or INCR16 type. But when such a burst is responded with SPLIT/RETRY/early burst termination, the AHB master will re-initiate the pending transfers of the burst with INCR or SINGLE burst-length type. It will terminate such INCR bursts when the original requested fixed-burst is transferred. In Fixed Burst-Length mode, if the DMA requests a burst transfer that is not equal to INCR4/8/16, the AHB Host interface splits the transfer into multiple burst transactions. For example, if the DMA requests a 15-beat burst transfer, the AHB interface splits it into multiple transfers of INCR8 and INCR4, and 3 SINGLE transactions.
- In unspecified burst mode, the AHB master will always initiate a transfer with INCR and complete the DMA requested burst in one go.
- In mixed burst mode, the AHB master will initiate bursts with fixed-size (INCRx) when the DMA requests transfers of size less than or equal to 16 beats. When DMA requests bursts of length more than 16, the AHB master will initiate such transfers with INCR and complete it in one go.
- Takes care of AHB SPLIT, RETRY, and ERROR conditions. Any ERROR response will halt all further transactions for that DMA, and indicate the error as fatal through the CSR and interrupt. The application must give a hard or software reset to the module to restart the operation.
- In any burst data transfer, the address bus value is always aligned to 32 bits(4byte) address size.
- The DMA Controller requests an AHB Burst Read transfer only when it can accept the received burst data completely.
- The DMA requests an AHB Burst Write transfer only when it has the sufficient data to transfer the burst completely. The AHB interface always assumes that it has data available to push into the AHB bus. However, the DMA can prematurely indicate end-of-valid data (due to the transfer of end-of-frame of an Ethernet frame) during the burst. In Fixed Burst Length mode, the AHB Master interface continues the burst with dummy data until the specified length is completed. In INCR mode, it takes steps to end the burst transfer prematurely.

AHB Slave Interface

- Supports single and burst transfers
- Supports 32-bit, 16-bit, and 8-bit write/read transfers to the CSR; 32-bit access to the CSR are recommended to avoid any software synchronization problems.
- Generates OKAY response only; does not generate ERROR responses.

3.3 DMA Controller

This section explains the DMA controller of the Ethernet-MAC. The DMA controller has independent Transmit and Receive engines, and a CSR (Control and Status registers) space. The Transmit Engine transfers data from system memory to the device port (MTL), while the Receive Engine transfers data from the device port to system memory. The DMA controller uses descriptors to efficiently move data from source to destination with minimal Host CPU intervention. The DMA is designed for packet-oriented data transfers such as frames in Ethernet. The controller can be programmed to interrupt the Host CPU for situations such as Frame Transmit and Receive transfer completion, and other normal/error conditions.

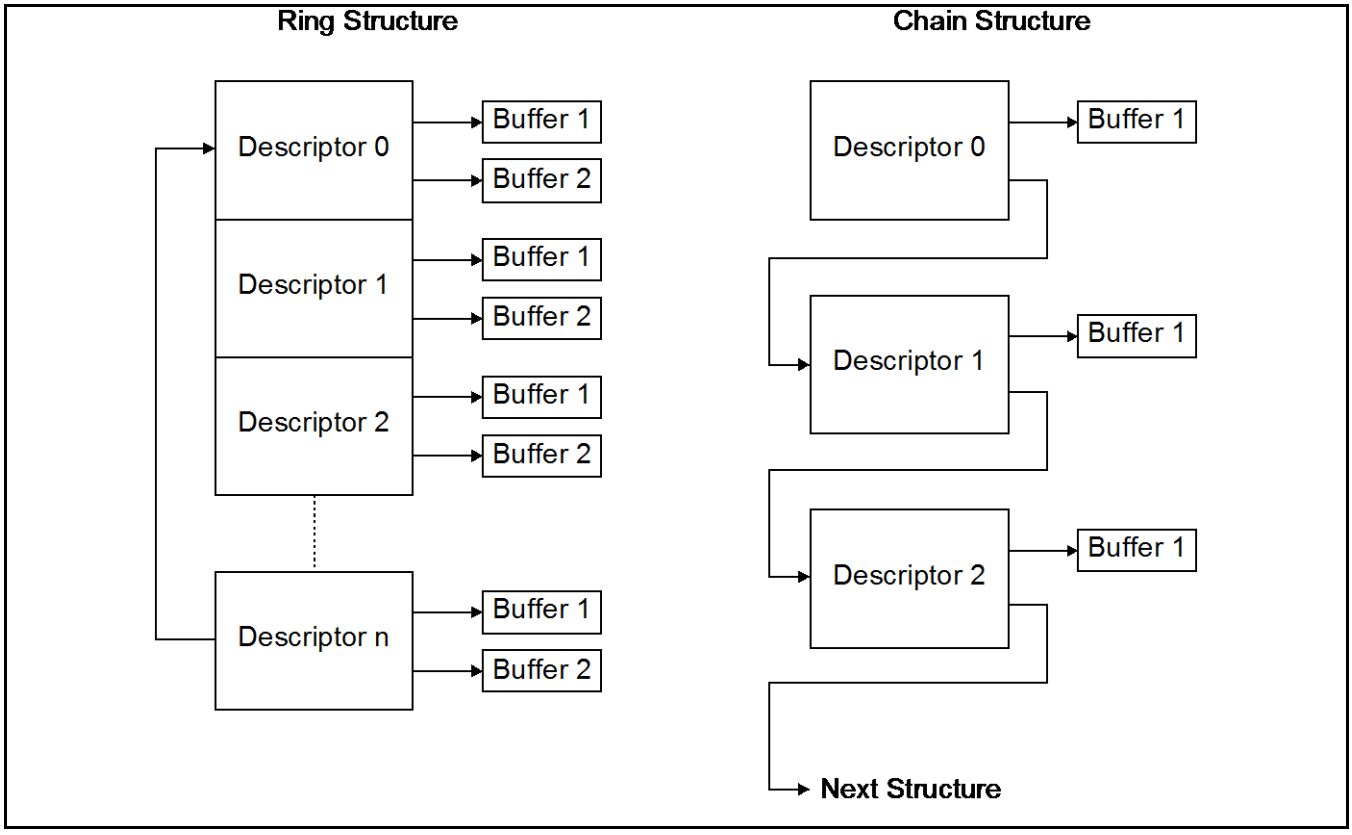
3.3.1 Descriptor Structure

The DMA and the Host driver communicate through two data structures:

- Control and Status registers (CSR)
- Descriptor lists and data buffers

Control and Status registers are described in detail in "4. Registers". Descriptors are described in detail in "5. Descriptors". The DMA transfers data frames received by the GMAC to the Receive Buffer in the Host memory, and Transmit data frames from the Transmit Buffer in the Host memory. Descriptors that reside in the Host memory act as pointers to these buffers.

Figure 3-1 Descriptor Ring and Chain Structure



There are two descriptor lists; one for reception, and one for transmission. The base address of each list is written into DMA registers 3 and 4, respectively. A descriptor list is forward linked (either implicitly or explicitly). The last descriptor may point back to the first entry to create a ring structure. Explicit chaining of descriptors is accomplished by setting the second address chained in both Receive and Transmit descriptors. Figure 3-1 shows the descriptor Ring and Chain structure. The descriptor lists resides in the Host physical memory address space. Each descriptor can point to a maximum of two buffers. This

enables two buffers to be used, physically addressed, rather than contiguous buffers in memory.
A data buffer resides in the Host physical memory space, and consists of an entire frame or part of a frame, but cannot exceed a single frame. Buffers contain only data. Buffer status is maintained in the descriptor. Data chaining refers to frames that span multiple data buffers. However, a single descriptor cannot span multiple frames. The DMA will skip to the next frame buffer when end-of-frame is detected. Data chaining can be enabled or disabled.
The descriptor ring and chain structure is shown in Figure 3-1.

3.3.2 Initialization of DMA Controller

Initialization for the Ethernet-MAC is as follows.

1. Write to GMAC register 0 to configure the operating mode and enable the transmit operation. The PS and DM bits are set based on the auto-negotiation result (read from the PHY).
2. Write to DMA register 0 to set Host bus access parameters.
3. Write to DMA register 7 to mask unnecessary interrupt causes.
4. The software driver creates the Transmit and Receive descriptor lists. Then it writes to both DMA register 3 and DMA register 4, providing the DMA with the starting address of each list.
5. Write to GMAC registers 1, 2, and 3 for desired filtering options.
6. Write to DMA register 6 to set bits 13 and 1 to start transmission and reception.
7. Write to GMAC register 0 to enable the Receive operation (bit 2: Receiver Enable).
   The Transmit and Receive engines enter the Running state and attempt to acquire descriptors from the respective descriptor lists. The Receive and Transmit engines then begin processing Receive and Transmit operations. The Transmit and Receive processes are independent of each other and can be started or stopped separately.

Host Bus Burst Access

The DMA will attempt to execute fixed-length Burst transfers on the AHB Master interface if configured to do so (FB bit of DMA register 0). The maximum Burst length is indicated and limited by the PBL field (DMA register 0[13:8]). The Receive and Transmit descriptors are always accessed in the maximum possible (limited by PBL or 16 * 8/bus width) burst-size for the 16-bytes to be read.
The Transmit DMA will initiate a data transfer only when sufficient space to accommodate the configured burst is available in MTL Transmit FIFO or the number of bytes till the end of frame (when it is less than the configured burst-length). The DMA will indicate the start address and the number of transfers required to the AHB Master Interface. When the AHB Interface is configured for fixed-length burst, then it will transfer data using the best combination of INCR4/8/16 and SINGLE transactions. Otherwise (no fixed-length burst), it will transfer data using INCR (undefined length) and SINGLE transactions.
The Receive DMA will initiate a data transfer only when sufficient data to accommodate the configured burst is available in MTL Receive FIFO or when the end of frame (when it is less than the configured burst length) is detected in the Receive FIFO. The DMA will indicate the start address and the number of transfers required to the AHB Master Interface. When the AHB Interface is configured for fixed-length burst, then it will transfer data using the best combination of INCR4/8/16 and SINGLE transactions. Otherwise (FB bit of DMA register 0 is reset), it will transfer data using INCR (undefined length) and SINGLE transactions.

Host Data Buffer Alignment

The Transmit and Receive data buffers do not have any restrictions on start address alignment. For example, in systems with 32-bit memory, the start address for the buffers can be aligned to any of the four bytes. However, the DMA always initiates transfers with address aligned to the 32-bit width with dummy data for the byte lanes not required. This typically happens during the transfer of the beginning or end of an Ethernet frame.

< Example 3-1 Buffer Read >

If the Transmit buffer address is 0x0FF2, and 15 bytes need to be transferred, then the DMA will read five 32-bit data from address 0x0FF0. But when transferring data to the MTL Transmit FIFO, the extra bytes (the first two bytes) will be dropped or ignored. Similarly, the last 3 bytes of the last transfer will also be ignored. The DMA always ensures it transfers a full 32-bit data to the MTL Transmit FIFO, unless it is the end-of-frame.

< Example 3-2 Buffer Write >

If the Receive buffer address is 0x0FF2 and 16 bytes of a received frame need to be transferred, then the DMA will write 5 32-bit data from address 0x0FF0. But the first 2 bytes of first transfer and the last 2 bytes of the fifth transfer will have dummy data.

Buffer Size Calculations

The DMA does not update the size fields in the Transmit and Receive descriptors. The DMA updates only the status fields (RDES and TDES) of the descriptors. The driver has to perform the size calculations.

The transmit DMA transfers the exact number of bytes (indicated by buffer size field of TDES1) towards the GMAC. If a descriptor is marked as first (FS bit of TDES0 is set), then the DMA marks the first transfer from the buffer as the start of frame. If a descriptor is marked as last (LS bit of TDES0 is set), then the DMA marks the last transfer from that data buffer as the end of frame to the MTL.

The Receive DMA transfers data to a buffer until the buffer is full or the end-of frame is received from the MTL. If a descriptor is not marked as last (LS bit of RDES0), then the descriptor's corresponding buffer(s) are full and the amount of valid data in a buffer is accurately indicated by its buffer size field minus the data buffer pointer offset when the FS bit of that descriptor is set. The offset is zero when the data buffer pointer is aligned to the data bus width. If a descriptor is marked as last, then the buffer may not be full (as indicated by the buffer size in RDES1). To compute the amount of valid data in this final buffer, the driver must read the frame length (FL bits of RDES0[29:16]) and subtract the sum of the buffer sizes of the preceding buffers in this frame. The Receive DMA always transfers next frame with a new descriptor.

Note:

- *Even when the start address of a receive buffer is not aligned to the system bus's data width, the system should allocate a receive buffer of a size aligned to the system bus width. For example, if the system allocates a 1,024-byte (1 KB) receive buffer starting from address 0x1000, the software can program the buffer start address in the Receive descriptor to have a 0x1002 offset. The Receive DMA writes the frame to this buffer with dummy data in the first two locations (0x1000 and 0x1001). The actual frame is written from location 0x1002. Thus, the actual useful space in this buffer is 1,022 bytes, even though the buffer size is programmed as 1,024 bytes, due to the start address offset.*

DMA Arbiter

The arbiter inside the DMA module performs the arbitration between the Transmit and Receive channel accesses to the AHB Master interface. Two types of arbitrations are possible: round-robin, and fixed-priority. When round-robin arbitration is selected (DA bit of DMA register 0 is reset), the arbiter allocates the data bus in the ratio set by the PR bits of DMA register 0, when both Transmit and Receive DMAs are requesting for access simultaneously. When the DA bit is set, the Receive DMA always gets priority over the Transmit DMA for data access. When the TXPR bit (bit 27 of DMA register 0) is also set, then the Transmit DMA gets priority over the Receive DMA.

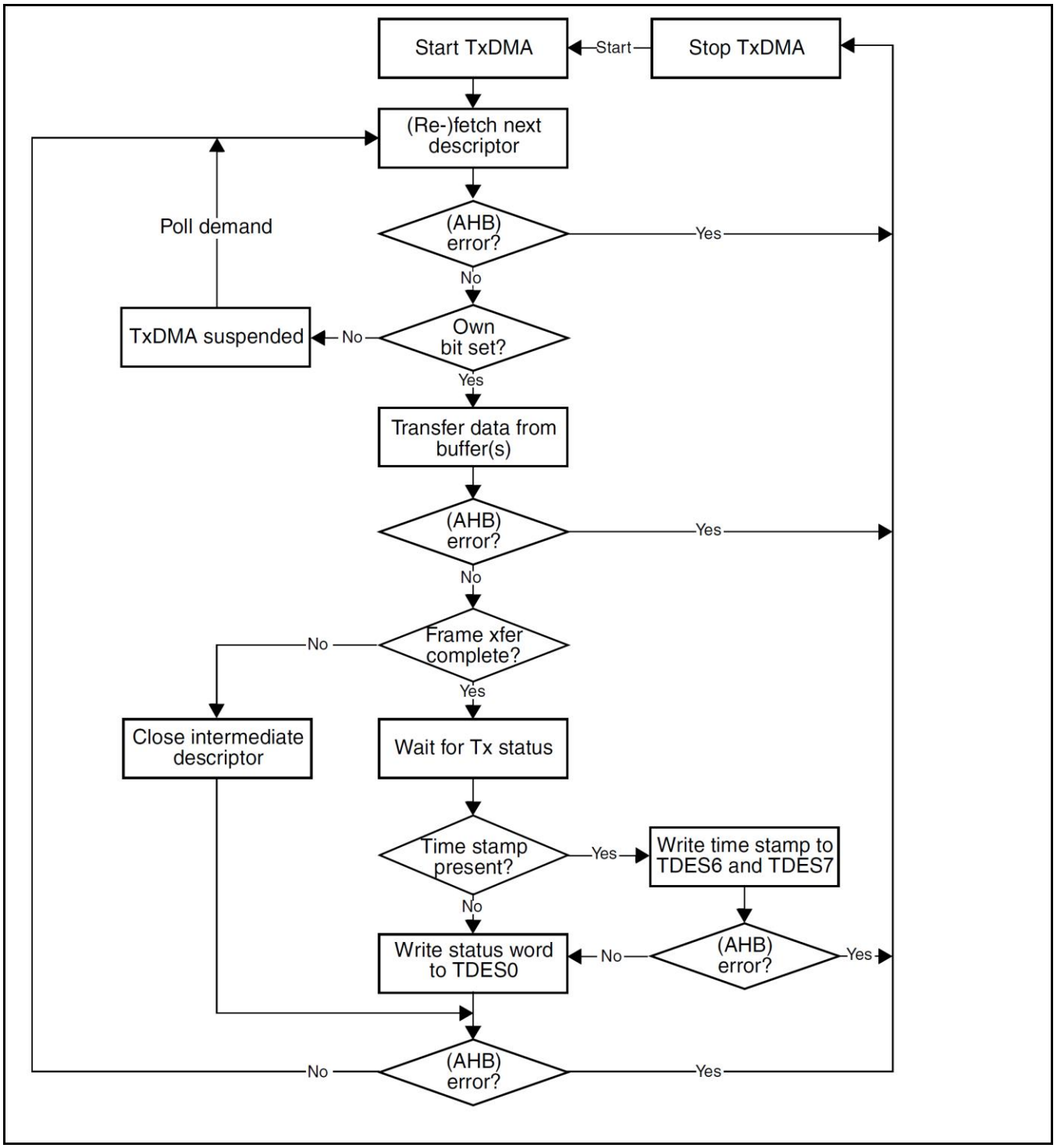
3.3.3 Transmission

Transmit DMA Operation: Default (Non-OSF) Mode

The Transmit DMA engine in default mode proceeds as follows: The TxDMA transmission flow in default mode is shown in Figure 3-2.

1. The Host sets up the transmit descriptor (TDES0-TDES3) and sets the OWN bit (TDES0[31]) after setting up the corresponding data buffer(s) with Ethernet Frame data.
2. Once the ST bit (DMA register 6[13]) is set, the DMA enters the Run state.
3. While in the Run state, the DMA polls the Transmit Descriptor list for frames requiring transmission. After polling starts, it continues in either sequential descriptor ring order or chained order. If the DMA detects a descriptor flagged as owned by the Host, or if an error condition occurs, transmission is suspended and both the Transmit Buffer Unavailable (DMA register 5[2]) and Normal Interrupt Summary (DMA register 5[16]) bits are set. The Transmit Engine proceeds to Step 9.
4. If the acquired descriptor is flagged as owned by DMA (TDES0[31] = 1), the DMA decodes the Transmit Data Buffer address from the acquired descriptor.
5. The DMA fetches the Transmit data from the Host memory and transfers the data to the MTL for transmission.
6. If an Ethernet frame is stored over data buffers in multiple descriptors, the DMA closes the intermediate descriptor and fetches the next descriptor. Steps 3, 4, and 5 are repeated until the end-of-Ethernet-frame data is transferred to the MTL.
7. When frame transmission is complete, if IEEE 1588 time stamping was enabled for the frame (as indicated in the transmit status) the time-stamp value obtained from MTL is written to the transmit descriptor (TDES6 and TDES7) that contains the end-of-frame buffer. The status information is then written to this transmit descriptor (TDES0). Because the OWN bit is cleared during this step, the Host now owns this descriptor. If time stamping was not enabled for this frame, the DMA does not alter the contents of TDES6 and TDES7.
8. Transmit Interrupt (DMA register 5[0]) is set after completing transmission of a frame that has Interrupt on Completion (TDES0[30]) set in its Last Descriptor. The DMA engine then returns to Step 3.
9. In the Suspend state, the DMA tries to re-acquire the descriptor (and thereby return to Step 3) when it receives a Transmit Poll demand and the Underflow Interrupt Status bit is cleared.

Figure 3-2 Transmit DMA Operation in Default Mode



Transmit DMA Operation: OSF Mode

While in the Run state, the transmit process can simultaneously acquire two frames without closing the Status descriptor of the first frame (if the OSF bit is set in DMA register 6[2]). As the transmit process finishes transferring the first frame, it immediately polls the Transmit Descriptor list for the second frame. If the second frame is valid, the transmit process transfers this frame before writing the first frame's status information.

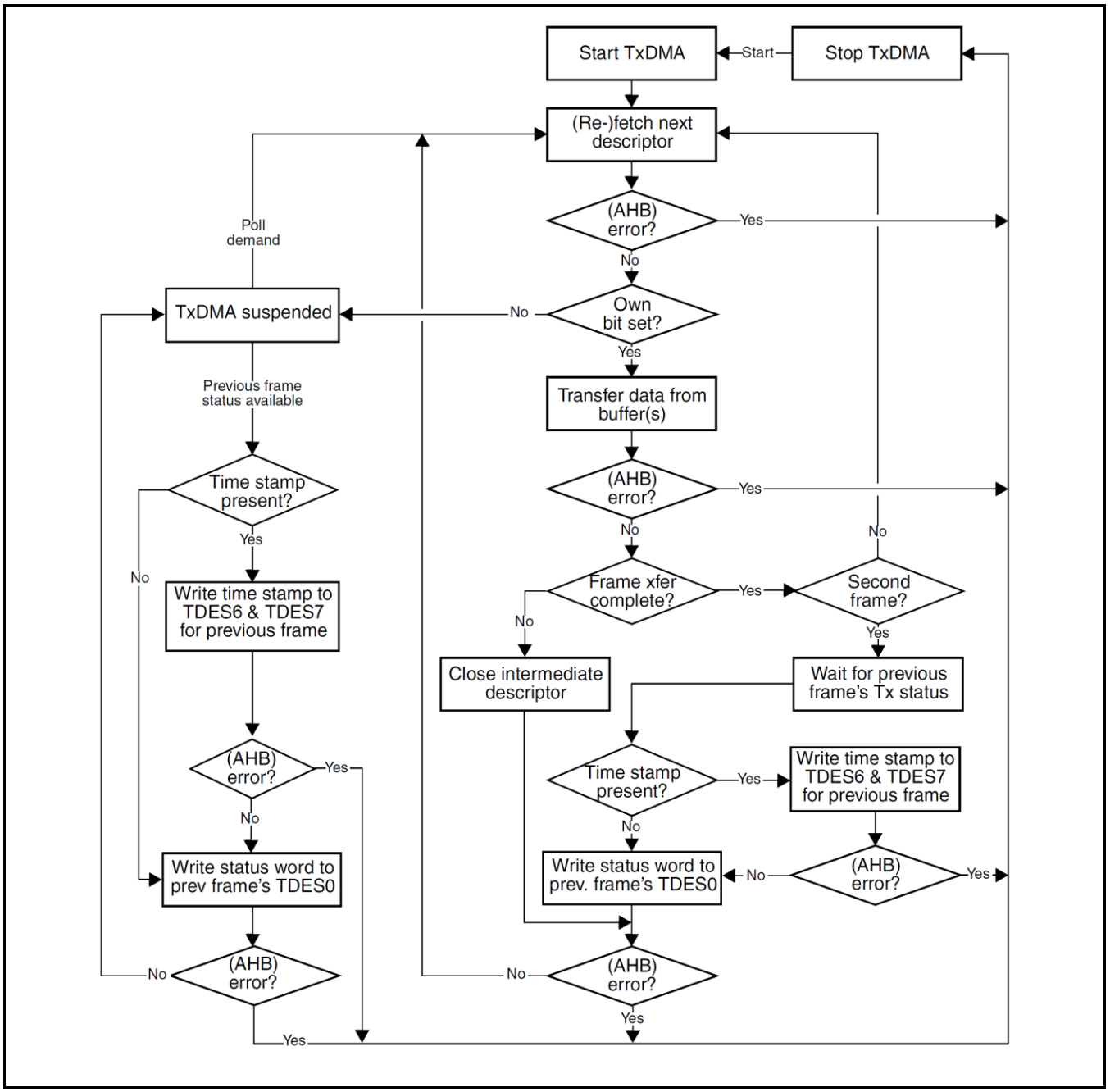
In OSF mode, the Run state Transmit DMA operates in the following sequence:

1. The DMA operates as described in steps 1–6 of the transmit DMA (default mode).
2. Without closing the previous frame's last descriptor, the DMA fetches the next descriptor.
3. If the DMA owns the acquired descriptor, the DMA decodes the transmit buffer address in this descriptor. If the DMA does not own the descriptor, the DMA goes into Suspend mode and skips to Step 7.
4. The DMA fetches the Transmit frame from the Host memory and transfers the frame to the MTL until the End-of-Frame data is transferred, closing the intermediate descriptors if this frame is split across multiple descriptors.
5. The DMA waits for the previous frame's frame transmission status and time stamp. Once the status is available, the DMA writes the time stamp to TDES6 and TDES7, if such time stamp was captured (as indicated by a status bit). The DMA then writes the status, with a cleared OWN bit, to the corresponding TDES0, thus closing the descriptor. If time stamping was not enabled for the previous frame, the DMA does not alter the contents of TDES6 and TDES7.
6. If enabled, the Transmit interrupt is set, the DMA fetches the next descriptor, and then proceeds to Step 3 (when Status is normal). If the previous transmission status shows an underflow error, the DMA goes into Suspend mode (Step 7).
7. In Suspend mode, if a pending status and time stamp are received from the MTL, the DMA writes the time stamp (if enabled for the current frame) to TDES6 and TDES7, then writes the status to the corresponding TDES0. It then sets relevant interrupts and returns to Suspend mode.
8. The DMA can exit Suspend mode and enter the Run state (go to Step 1 or Step 2 depending on pending status) only after receiving a Transmit Poll demand (DMA register 1).

Note:

- *As the DMA fetches the next descriptor in advance before closing the current descriptor, the descriptor chain should have more than 3 different descriptors for correct and proper operation.*

Figure 3-3 Transmit DMA Operation in OSF Mode



Transmit Frame Processing

The Transmit DMA expects that the data buffers contain complete Ethernet frames, excluding preamble, pad bytes, and FCS fields. The DA, SA, and Type/Len fields contain valid data. If the Transmit Descriptor indicates that the GMAC must disable CRC or PAD insertion, the buffer must have complete Ethernet frames (excluding preamble), including the CRC bytes. Frames can be data-chained and can span several buffers. Frames must be delimited by the FS(TDES0[28]) and the LS(TDES0[29]), respectively. As transmission starts, the First Descriptor must have FS(TDES0[28]) set. When this occurs, frame data transfers from the Host buffer to the MTL Transmit FIFO. Concurrently, if the current frame has the

LS(TDES0[29]) clear, the Transmit Process attempts to acquire the Next Descriptor. The Transmit Process expects this descriptor to have FS(TDES0[28]) clear. If LS(TDES0[29]) is clear, it indicates an intermediary buffer. If LS(TDES0[29]) is set, it indicates the last buffer of the frame. After the last buffer of the frame has been transmitted, the DMA writes back the final status information to the TDES0 word of the descriptor that has TDES1[29] is set. At this time, if Interrupt on Completion (TDES0[30]) was set, Transmit Interrupt (DMA register 5[0]) is set, the Next Descriptor is fetched, and the process repeats. Actual frame transmission begins after the MTL Transmit FIFO has reached either a programmable transmit threshold (DMA register 6[16:14]), or a full frame is contained in the FIFO. There is also an option for Store and Forward Mode (DMA register 6[21]). Descriptors are released (OWN bit TDES0[31] clears) when the DMA finishes transferring the frame from the PHY interface.

Transmit Polling Suspended

Transmit polling can be suspended by either of the following conditions:

- The DMA detects a descriptor owned by the Host (TDES0[31]=0). To resume, the driver must give descriptor ownership to the DMA and then issue a Poll Demand command.
- A frame transmission is aborted when a transmit error due to underflow is detected. The appropriate Transmit Descriptor 0 (TDES0) bit is set.

If the second condition occurs, both Abnormal Interrupt Summary (DMA register 5[15]) and Transmit Underflow bits (DMA register 5 [5]) are set, and the information is written to Transmit Descriptor 0, causing the suspension. If the DMA goes into the Suspend state due to the first condition, then both Normal Interrupt Summary (DMA register 5 [16]) and Transmit Buffer Unavailable (DMA register 5 [2]) are set.
In both cases, the position in the Transmit List is retained. The retained position is that of the descriptor following the Last Descriptor closed by the DMA.
The driver must explicitly issue a Transmit Poll Demand command after rectifying the suspension cause.

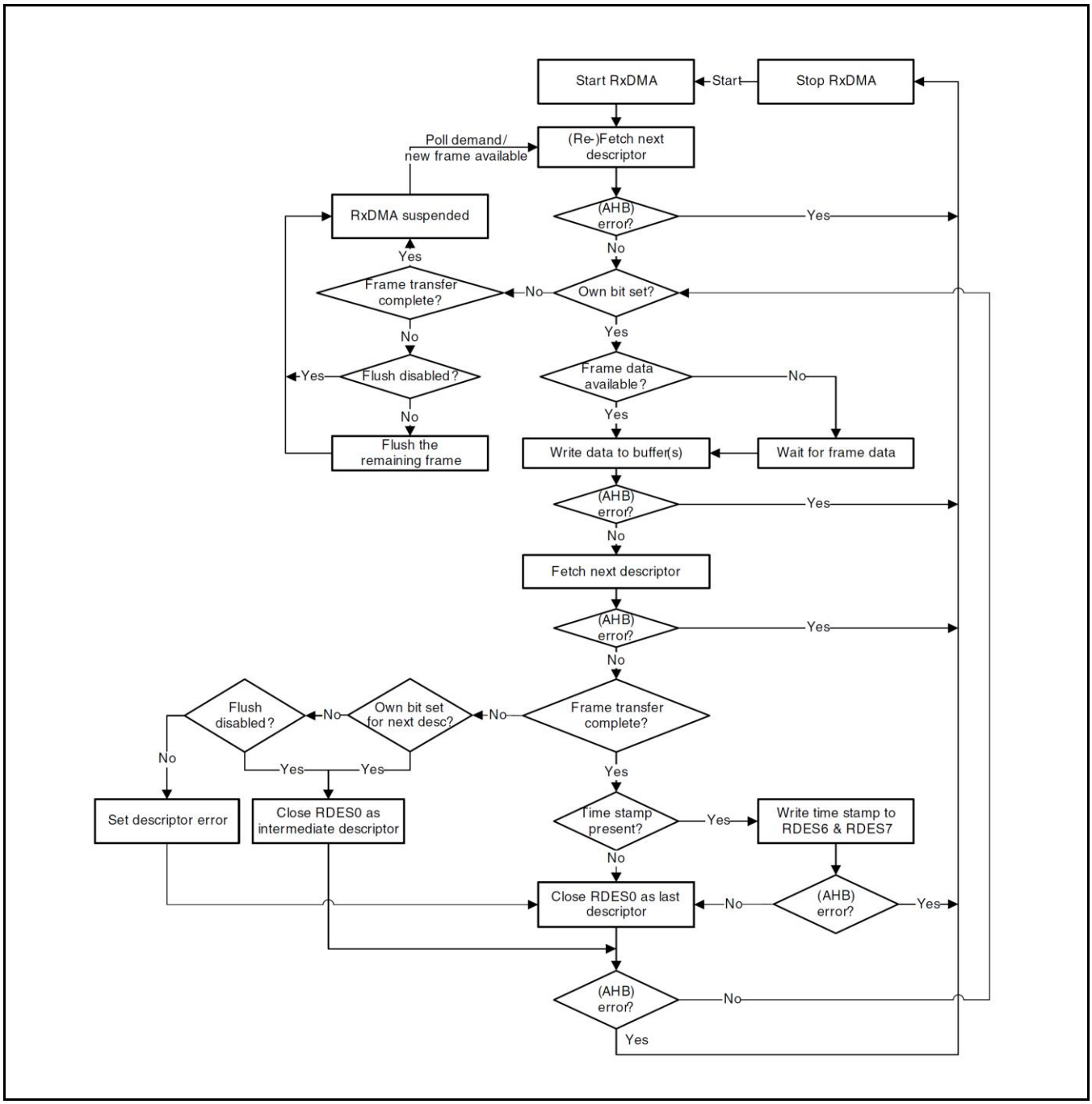
3.3.4 Reception

The Receive DMA engine's reception sequence is depicted in Figure 3-4 and proceeds as follows:

1. The host sets up Receive descriptors (RDES0-RDES3) and sets the OWN bit (RDES0[31]).
2. Once the SR (DMA register 6[1]) bit is set, the DMA enters the Run state. While in the Run state, the DMA polls the Receive Descriptor list, attempting to acquire free descriptors. If the fetched descriptor is not free (is owned by the host), the DMA enters the Suspend state and jumps to Step 9.
3. The DMA decodes the receive data buffer address from the acquired descriptors.
4. Incoming frames are processed and placed in the acquired descriptor's data buffers.
5. When the buffer is full or the frame transfer is complete, the Receive engine fetches the next descriptor.
6. If the current frame transfer is complete, the DMA proceeds to Step 7. If the DMA does not own the next fetched descriptor and the frame transfer is not complete (EOF is not yet transferred), the DMA sets the Descriptor Error bit in the RDES0 (unless flushing is disabled). The DMA closes the current descriptor (clears the OWN bit) and marks it as intermediate by clearing the Last Segment (LS) bit in the RDES0 value (marks it as Last Descriptor if flushing is not disabled), then proceeds to Step 8. If the DMA does own the next descriptor but the current frame transfer is not complete, the DMA closes the current descriptor as intermediate and reverts to Step 4.
7. If IEEE 1588 time stamping is enabled, the DMA writes the time stamp (if available) to the current descriptor's RDES6 and RDES7. It then takes the receive frame's status from the MTL and writes the status word to the current descriptor's RDES0, with the OWN bit cleared and the Last Segment bit set.
8. The Receive engine checks the latest descriptor's OWN bit. If the host owns the descriptor (the OWN bit is 0) the Receive Buffer Unavailable bit (register 5[7]) is set and the DMA Receive engine enters the Suspended state (Step 9). If the DMA owns the descriptor, the engine returns to Step 4 and awaits the next frame.
9. Before the Receive engine enters the Suspend state, partial frames are flushed from the Receive FIFO (You can control flushing using bit24 of DMA register 6).
10. The Receive DMA exits the Suspend state when a Receive Poll demand is given or the start of next frame is available from the MTL's Receive FIFO. The engine proceeds to Step 2 and refetches the next descriptor.

The DMA does not acknowledge accepting the status from the MTL until it has completed the time stamp write-back and is ready to perform status write-back to the descriptor.
If software has enabled time stamping through CSR, when a valid time stamp value is not available for the frame (for example, because the receive FIFO was full before the time stamp could be written to it), the DMA writes all-ones to RDES6 and RDES7. Otherwise (that is, if time stamping is not enabled), the RDES6 and RDES7 remain unchanged.

Figure 3-4 Receive DMA Operation



Receive Descriptor Acquisition

The Receive Engine always attempts to acquire an extra descriptor in anticipation of an incoming frame. Descriptor acquisition is attempted if any of the following conditions is satisfied:

- The receive Start/Stop bit (DMA register 6[1]) has been set immediately after being placed in the Run state.
- The data buffer of current descriptor is full before the frame ends for the current transfer.
- The controller has completed frame reception, but the current Receive Descriptor is not yet closed.
- The receive process has been suspended because of a host-owned buffer (RDES0[31] = 0) and a new frame is received.
- A Receive poll demand has been issued.

Receive Frame Processing

The GMAC transfers the received frames to the Host memory only when the frame passes the address filter and frame size is greater than or equal to configurable threshold bytes set for the Receive FIFO of MTL, or when the complete frame is written to the FIFO in Store-and-Forward mode. If the frame fails the address filtering, it is dropped in the GMAC block itself (unless Receive All bit in GMAC register 1[31] is set). Frames that are shorter than 64 bytes, because of collision or premature termination, can be purged from the MTL Receive FIFO. After 64 (configurable threshold) bytes have been received, the MTL block requests the DMA block to begin transferring the frame data to the Receive Buffer pointed to by the current descriptor. The DMA sets First Descriptor (RDES0[9]) after the AHB Interface becomes ready to receive a data transfer (if DMA is not fetching transmit data from the host), to delimit the frame. The descriptors are released when the OWN (RDES0[31]) bit is reset to 0, either as the Data buffer fills up or as the last segment of the frame is transferred to the Receive buffer. If the frame is contained in a single descriptor, both LS (RDES0[8]) and FS (RDES0[9]) are set. The DMA fetches the next descriptor, sets the LS (RDES0[8]) bit, and releases the RDES0 status bits in the previous frame descriptor. Then the DMA sets Receive Interrupt (DMA register 5[6]). The same process repeats unless the DMA encounters a descriptor flagged as being owned by the host. If this occurs, the Receive Process sets Receive Buffer Unavailable (DMA register 5[7]) and then enters the Suspend state. The position in the receive list is retained.

Receive Process Suspended

If a new Receive frame arrives while the Receive Process is in Suspend state, the DMA refetches the current descriptor in the Host memory. If the descriptor is now owned by the DMA, the Receive Process re-enters the Run state and starts frame reception. If the descriptor is still owned by the host, by default, the DMA discards the current frame at the top of the MTL receive FIFO and increments the missed frame counter. If more than one frame is stored in the MTL receive FIFO, the process repeats. The discarding or flushing of the frame at the top of the MTL receive FIFO can be avoided by setting Operation Mode register bit24 (DFF). In such conditions, the receive process sets the Receive Buffer Unavailable status and returns to the Suspend state.

3.3.5 Interrupts

Interrupts can be generated as a result of various events. DMA register 5 contains all the bits that might cause an interrupt. DMA register 7 contains an enable bit for each of the events that can cause an interrupt. There are two groups of interrupts, Normal and Abnormal, as described in DMA register 5. Interrupts are cleared by writing 1 to the corresponding bit position. When all the enabled interrupts within a group are cleared, the corresponding summary bit is cleared. When both the summary bits are cleared, the interrupt signal INT_SBD is deasserted. If the Ethernet-MAC is the cause for assertion of the interrupt, then any of the GLI, GMI, or GPI bits of DMA register 5 will be set to 1.

Note:

- *DMA register 5 is the (interrupt) status register. The interrupt signal (INT_SBD) will be asserted due to any event in this status register only if the corresponding interrupt enable bit is set in DMA register 7.*

Interrupts are not queued and if the interrupt event occurs before the driver has responded to it, no additional interrupts are generated. For example, Receive Interrupt (DMA register 5[6]) indicates that one or more frames were transferred to the Host buffer. The driver must scan all descriptors, from the last recorded position to the first one owned by the DMA.
An interrupt is generated only once for simultaneous, multiple events. The driver must scan DMA register 5 for the cause of the interrupt. The interrupt is not generated again unless a new interrupting event occurs, after the driver has cleared the appropriate bit in DMA register 5. For example, the controller generates a Receive interrupt (DMA register 5[6]) and the driver begins reading DMA register 5. Next, Receive Buffer Unavailable (DMA register 5[7]) occurs. The driver clears the Receive interrupt. Even then, the INT_SBD signal is not deasserted, due to the active or pending Receive Buffer Unavailable interrupt.
An interrupt timer (see "4.42. DMA Register 9 (RIWTR)") is given for flexible control of Receive Interrupt (DMA register 5[6]). When this Interrupt timer is programmed with a non-zero value, it will get activated as soon as the RxDMA completes a transfer of a received frame to system memory without asserting the Receive Interrupt because it is not enabled in the corresponding Receive Descriptor (RDES1[31]). When this timer runs out as per the programmed value, RI bit is set and the interrupt is asserted if the corresponding RI is enabled in DMA register 7. This timer gets disabled before it runs out, when a frame is transferred to memory and the RI is set because it is enabled for that descriptor.

3.3.6 Error Response to DMA

For any data transfer initiated by a DMA channel, if the slave replies with an error response, that DMA stops all operations and updates the error bits and the Fatal Bus Error bit in the Status register (DMA register 5). That DMA controller can resume operation only after software resetting or hardware resetting and re-initializing the DMA.

3.3.7 CRC

DMA generate CRC for the FCS field of the Ethernet frame.
DMA calculates the 32-bit CRC for the FCS field of the Ethernet frame. The encoding is defined by the following generating polynomial.

G (x) = x32 + x26 + x23 + x22 + x16 + x12 + x11 + x10 + x8 + x7 + x5 + x4 + x2 + x + 1

3.4 Checksum Engine

Communication protocols such as TCP and UDP implement checksum fields, which help determine the integrity of data transmitted over a network. Because the most widespread use of Ethernet is to encapsulate TCP and UDP over IP datagrams, the GMAC has an optional Checksum Offload Engine (COE) to support checksum calculation and insertion in the transmit path, and error detection in the receive path.

3.4.1 Transmit Checksum Offload Engine

The transmit checksum offload engine (COE) supports two types of checksum calculation and insertion. This checksum engine can be controlled for each frame by setting the CIC bits (TDES0[23:22]).

IP Header Checksum Engine

In IPv4 datagrams, the integrity of the header fields is indicated by the 16-bit Header Checksum field (the eleventh and twelfth bytes of the IPv4 datagram). The COE detects an IPv4 datagram when the Ethernet frame's Type field has the value 0x0800 and the IP datagram's Version field has the value 0x4. At the transmission, the frame's checksum field is ignored during calculation and replaced with the calculated value. IPv6 headers do not have a checksum field; thus, the COE does not modify IPv6 header fields. The result of this IP header checksum calculation is indicated by the IP Header Error status bit in the Transmit status (TDES0[16]). This status bit is set whenever the values of the Ethernet frame Type field and the IP header's Version field are not consistent, or when the Ethernet frame does not have enough data, as indicated by the IP header Length field.
In other words, this bit is set when an IP header error is asserted under the following circumstances:

For IPv4 datagrams:

- The Ethernet type is 0x0800, but the IP header's Version field does not equal 0x4
- The IPv4 Header Length field indicates a value less than 0x5 (20 bytes)
- The total frame length is less than the value given in the IPv4 Header Length field

For IPv6 datagrams:

- The Ethernet type is 0x86DD but the IP header Version field does not equal 0x6
- The frame ends before the IPv6 header (40 bytes) or extension header (as given in the corresponding Header Length field in an extension header) is completely received.

Even when the COE detects such an IP header error, it inserts an IPv4 header checksum if the Ethernet frame Type field indicates an IPv4 payload.

TCP/UDP/ICMP Checksum Engine

The TCP/UDP/ICMP Checksum Engine processes the IPv4 or IPv6 header (including extension headers) and determines whether the encapsulated payload is TCP, UDP, or ICMP.

Notes:

- *For non-TCP, -UDP, or -ICMP/ICMPv6 payloads, this checksum engine is bypassed and nothing further is modified in the frame.*
  Fragmented IP frames (IPv4 or IPv6) and IP frames with security features (such as an authentication header or encapsulated security payload) are not processed by this engine, and therefore must be bypassed. In other words, payload checksum insertion must not be enabled for such frames.
  In Ethernet environment using IPv6 Authentication Header, COE must not be enabled.
  CIC(Transmit Enhanced Descriptor 0) must be 00 or 01.

The checksum is calculated for the TCP, UDP, or ICMP payload and inserted into its corresponding field in the header. This engine can work in the following two modes:

- In the first mode, the TCP, UDP, or ICMPv6 pseudo-header is not included in the checksum calculation and is assumed to be present in the input frame's Checksum field. This engine includes the Checksum field in the checksum calculation, then replaces the Checksum field with the final calculated checksum.
- In the second mode, the engine ignores the Checksum field, includes the TCP, UDP, or ICMPv6 pseudo-header data into the checksum calculation, and overwrites the checksum field with the final calculated value.

Note:

- *For ICMP-over-IPv4 packets, the Checksum field in the ICMP packet must always be 0x0000 in both modes, because pseudo-headers are not defined for such packets. If it does not equal 0x0000, an incorrect checksum may be inserted into the packet.*

The result of this operation is indicated by the IP Payload bit in TDES0[12]. This engine sets the IP Payload Error when it detects that the frame has been forwarded to the GMAC Transmitter engine in Store-and-Forward mode without the end-of-frame being written to the FIFO, or when the packet ends before the number of bytes indicated by the Payload Length field in the IP Header is received. When the packet is longer than the indicated payload length, the COE ignores them as stuff bytes, and no error is reported. When this engine detects the first type of error, it does not modify the TCP, UDP, or ICMP header. For the second error type, it still inserts the calculated checksum into the corresponding header field.

Note:

- *The checksum for TCP, UDP, or ICMP is calculated over a complete frame, then inserted into its corresponding header field. Due to this requirement, this function is enabled only when the Transmit FIFO is configured for Store-and-Forward mode (that is, when the TSF bit is set in DMA register 6). If the GMAC is configured for Threshold (cut-through) mode, the Transmit COE is bypassed.*
  Even in the store-and-forward mode, the checksum insertion for TCP, UDP, or ICMP must enable only in the frames that are less than the following number of bytes size.
  Frame length (Byte) < 2036(Byte) – PBL×4(Byte)
  The PBL is the burst length in the transmission specified by the DMA register 0.
  In the transmission, the allowable maximum value for the PBL×4 is 512 bytes. Therefore, the burst length is enough for the normal Ethernet frame.
  However, if the above frame length is not satisfied, the checksum insertion engine fails and consequently all succeeding frames may get corrupted due to improper recovery.

3.4.2 Receive Checksum Offload Engine

Both IPv4 and IPv6 frames in the received Ethernet frames are detected and processed for data integrity check. You can enable this module by setting the IPC bit in the GMAC Configuration register. The GMAC receiver identifies IPv4 or IPv6 frames by checking for value 0x0800 or 0x86DD, respectively, in the received Ethernet frames' Type field. This identification applies to VLAN-tagged frames as well. The Receive Checksum Offload engine calculates IPv4 header checksums and checks that they match the received IPv4 header checksums. The IP Header Error bit is set for any mismatch between the indicated payload type (Ethernet Type field) and the IP header version, or when the received frame does not have enough bytes, as indicated by the IPv4 header's Length field (or when fewer than 20 bytes are available in an IPv4 or IPv6 header). This engine also identifies a TCP, UDP or ICMP payload in the received IP datagrams (IPv4 or IPv6) and calculates the checksum of such payloads properly, as defined in the TCP, UDP, or ICMP specifications. This engine includes the TCP/UDP/ICMPv6 pseudo-header bytes for checksum calculation and checks whether the received checksum field matches the calculated value. The result of this operation is given as a Payload Checksum Error bit in the receive status word. This status bit is also set if the length of the TCP, UDP, or ICMP payload does not tally to the expected payload length given in the IP header. As mentioned in "TCP/UDP/ICMP Checksum Engine", this engine bypasses the payload of fragmented IP datagrams, IP datagrams with security features and payloads other than TCP, UDP or ICMP. This information (whether the checksum engine is bypassed or not) is given in the receive status.

Note:

- *This checksum engine is bypassed the receive packet and ignored when GMAC is received the packet with non-zero value in the Segment Left field in the IPv6 Routing extension header.*
  In Ethernet environment using IPv6 Authentication Header, COE must not be enabled.
  IPC(MCR(GMAC register0)-bit10) must be reset.

3.5 Energy Efficient Ethernet

Energy Efficient Ethernet (EEE) is an optional operational mode that enables the IEEE 802.3 Media Access Control (MAC) sublayer along with a family of Physical layers to operate in the Low-Power Idle (LPI) mode.
The LPI mode allows power saving by switching off parts of the communication device functionality when there is no data to be transmitted and received. The systems on both sides of the link can disable some functions and save power during the periods of low-link usage. The GMAC controls whether the system should enter or exit the LPI mode and communicates this to the PHY.
The EEE specifies the capabilities negotiation methods that the link partners can use to determine whether EEE is supported and then select the set of parameters that common to both devices.
The LPI mode is supported only in the full-duplex mode when GMAC uses MII.

Transmit Path Functions

In the transmit path, the software must set the bit 16 (LPIEN) of GMAC register 12 (LPI Control and Status register) to indicate to the GMAC to stop transmission and initiate the LPI protocol. The GMAC completes the transmission in progress, generates its transmission status, and then starts transmitting the LPI pattern instead of the IDLE pattern during Interframe gap (IFG).

To make the PHY enter the LPI state, the GMAC performs the following tasks:

1. De-asserts TX_EN.
2. Asserts TX_ER.
3. Sets TXD[3:0] to 0x1 (for 100 Mbps).
   *The GMAC maintains the same state of the TX_EN, TX_ER, and TXD signals for the entire duration during which the PHY remains in the LPI state.
4. Updates the status (bit0 of GMAC register 12 (LPI Control and Status register)) and generates an interrupt.

To bring the PHY out of the LPI state, that is, when the software resets the LPIEN bit, the GMAC performs the following tasks:

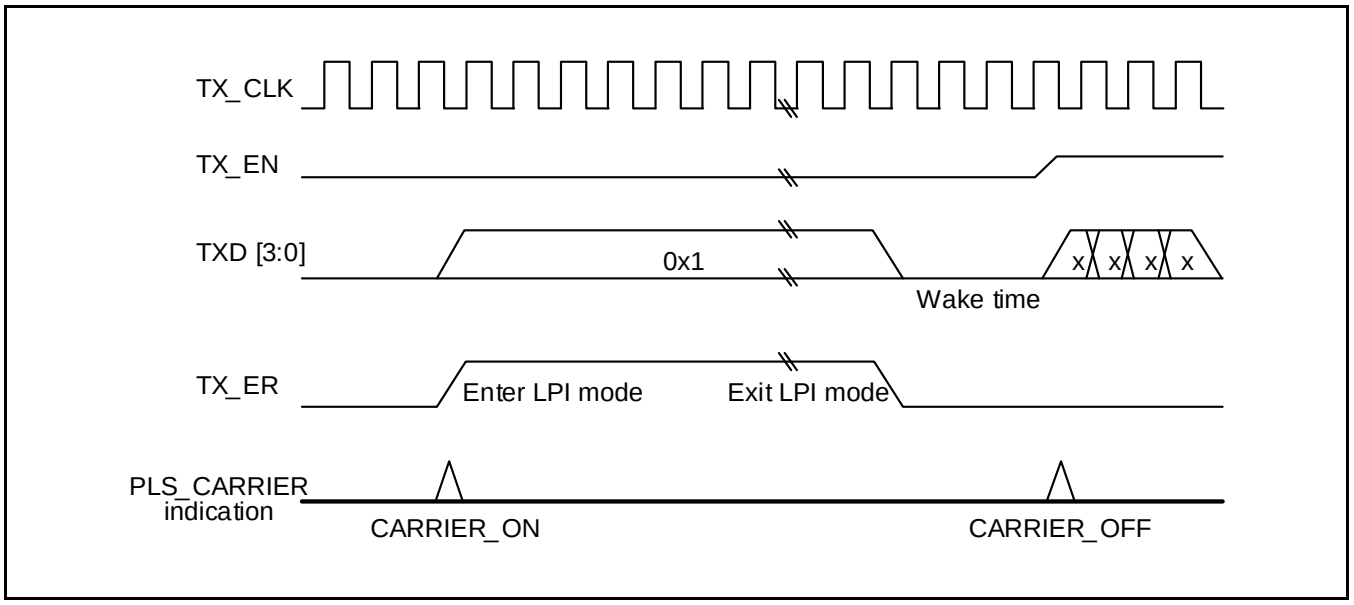
1. Stops transmitting the LPI pattern and starts transmitting the IDLE pattern.
2. Starts the LPI TW TIMER.
   The GMAC cannot start the transmission until the wake-up time specified for the PHY expires. The auto-negotiated wake-up interval is programmed in bits [15:0] (TWT: LPI TW TIMER) of GMAC register 13 (LPI Timers Control register).
3. Updates the LPI exit status (bit1 of GMAC register 12 (LPI Control and Status register)) and generates an interrupt.

Note:

- *When the TX clock gating is done during the LPI mode, you cannot use the bit19 (LPITXA) of GMAC register 12 (LPI Control and Status register).*
  The Energy Efficient Ethernet standard (802.3az) specifies the PHY cannot stop the TX_CLK during the LPI state in the MII(10 Mbps/100 Mbps) mode.
- *If the GMAC is in the transmit LPI mode and the supply of TX_CLK is stopped, the application should not write to CSR registers that are synchronized to TX_CLK domain.*
- *If the GMAC is in the LPI mode and the host issues a software-reset or hardware reset, the GMAC transmitter come out of the LPI mode.*

Figure 3-5 shows the behavior of TX_EN, TX_ER, and TXD[3:0] signals during the LPI mode transitions.

Figure 3-5 LPI Transitions (Transmit)



Receive Path Functions

In the receive path, when the PHY receives the signals from the link partner to enter into the LPI state, the PHY and GMAC perform the following tasks:

1. The PHY asserts RX_ER.
2. The PHY sets RXD[3:0] to 0x1.
3. The PHY de-asserts RX_DV.
   * The PHY maintains the same state of the RX_ER, RXD, and RX_DV signals for the entire duration during which it remains in the LPI state.
4. The GMAC updates the bit2 (RLPIEN) of GMAC register 12 (LPI Control and Status register) and generates an interrupt immediately.

When the PHY receives signals from the link partner to exit the LPI state, the PHY and GMAC perform the following tasks:

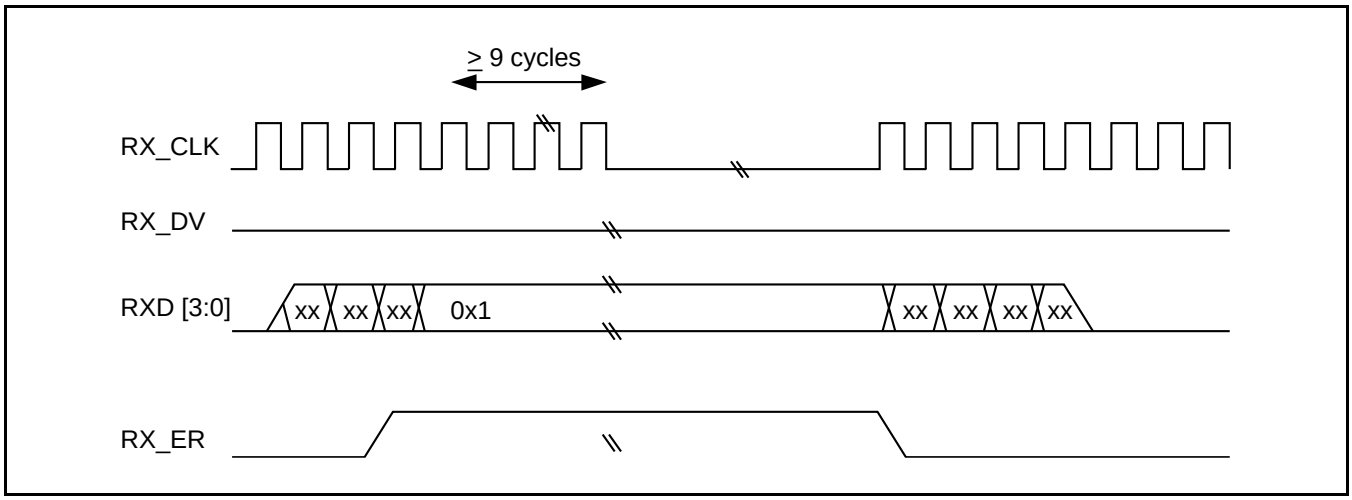
1. The PHY de-asserts RX_ER and returns to a normal inter-frame state.
2. The GMAC updates the bit3 (RLPIEX) of GMAC register 12 (LPI Control and Status register) and generates an interrupt immediately. In this case, the interrupt signal INT_LPI (synchronous to Rx clock) is also asserted.

Figure 3-6 shows the behavior of RX_ER, RX_DV, and RXD[3:0] signals during the LPI mode transitions.

Note:

- *If the RX_CLK_stoppable bit (in the PHY register written through MDIO) is asserted when the PHY is indicating LPI to the GMAC, then the PHY may halt the RX_CLK at any time more than 9 clock cycles after the start of the LPI state as shown in Figure 3-6.*
  If the GMAC is in the LPI mode and the host issues a software reset or hardware reset, the GMAC receiver comes out of the LPI mode during reset. If the LPI pattern is still received after the reset is de-asserted, the GMAC receiver again enters the LPI state.
  If the RX clock is stopped in the RX LPI mode, the host should not write to the CSR registers that are being synchronized to the RX clock domain.

Figure 3-6 LPI Transitions (Receive)



LPI Timers

The GMAC transmitter maintains the following two timers that are loaded with the respective values (from GMAC register 13 (LPI Timers Control register)):

■ LPI LS TIMER

The LPI LS TIMER counts, in milliseconds, the time expired since the link status is up.
This timer is cleared every time the link goes down and is incremented when the link is up again and the terminal count as programmed by the software is reached. The PHY interface does not assert the LPI pattern unless the terminal count is reached. This ensures a minimum time for which no LPI pattern is asserted after a link is established with the remote station. This period is defined as 1 second in the IEEE standard 802.3-az-2010. The LPI LS TIMER is 10-bit wide. Therefore, the software can program up to 1023 ms.

■ LPI TW TIMER

The LPI TW TIMER counts, in microseconds, the time expired since the de-assertion of LPI. The terminal count of the timer is the value of resolved Transmit TW that is the auto-negotiated time after which the GMAC can resume the normal transmit operation. The GMAC supports the LPI TWTIMER in units of microsecond. The LPI TW TIMER is 16-bit wide. Therefore, the software can program up to 65535 µs.

* Program the bit17 (PLS) of GMAC register 12 (LPI Control and Status register) to 0 when the supply of the TX_CLK is stopped from PHY. This resets the internal timers. If the mode is changed after the LPI LS TIMER or LPI TW TIMER starts, the change in the TX_CLK frequency can result in incorrect timeout.

3.6 MAC Management Counters

The MAC Management Counters (MMC) maintains a set of registers for gathering statistics on the received and transmitted frames. These include a control register for controlling the behavior of the registers, three 32-bit registers containing interrupts generated (receive and transmit), three 32-bit registers containing masks for the Interrupt register (receive and transmit), and a statistics counter register.

MMC Operation

These registers are accessible from the Application through the AHB slave interface. Each register is 32-bit wide. The word access (32-bit wide) is only acceptable. "Table 4-5 MMC register Map in 4.48 MMC register list" describe the functions(count content) of each statistics counter. The Receive MMC counters are updated for frames that are passed by the Address Filter (AFM) block. Statistics of frames that are dropped by the AFM module are not updated unless they are runt frames of less than 6 bytes (DA bytes are not received fully). The received IPC counter is updated only with respect to the frame that has passed through the destination address filter. The received IPC counter gathers statistics on encapsulated IPv4, IPv6, TCP, UDP, ICMP payloads in received Ethernet frames.

By the MMC control register, the operation mode of management counter can be set. The event that the counter reaches a predetermined value can be generated an interrupt. It can be checked the status of the interrupt by the next three registers.

- MMC receive interrupt register
- MMC transmit interrupt register
- MMC receive checksum offload interrupt register
  It can be masked individually each interrupt by the next registers.
- MMC receive interrupt mask register
- MMC transmit interrupt mask register
- MMC receive checksum offload interrupt mask register

The MMC register naming convention is as follows.

- "tx" as a prefix or suffix indicates counters associated with transmission
- "rx" as a prefix or suffix indicates counters associated with reception
- "_g" as a suffix indicates registers that count good frames only
- "_gb" as a suffix indicates registers that count frames regardless of whether they are good or bad

Transmitted frames are considered "good" if transmitted successfully. In other words, a transmitted frame is good if the frame transmission is not aborted due to any of the following errors:

- Jabber Timeout
- No Carrier/Loss of Carrier
- Late Collision
- Frame Underflow
- Excessive Deferral
- Excessive Collision

Received frames are considered "good" if none of the following errors exists:

- CRC error
- Runt Frame (shorter than 64 bytes)
- Alignment error (in 10/100 Mbps only)
- Length error (non-Type frames only)
- Out of Range (non-Type frames only, longer than maximum size)
- MII_RXER Input error

The maximum frame size depends on the frame type, as follows:

- Untagged frame maxsize = 1518
- VLAN Frame maxsize = 1522
- Jumbo Frame maxsize = 9018
- JumboVLAN Frame maxsize = 9022

3.7 Station Management Agent

The Station Management Agent (SMA) module allows the Application to access any PHY registers through a 2-wire Station Management interface (MIM). The interface supports accessing up to 32 PHYs. The application can select one of the 32 PHYs and one of the 32 registers within any PHY and send control data or receive status information.

Functions

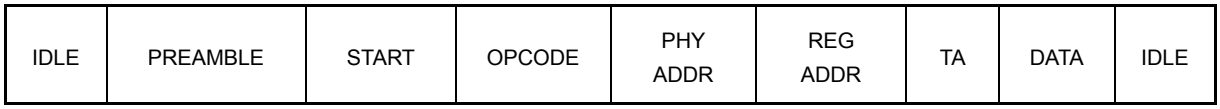
The MDC clock signal used in the management operation is a divided clock from SYS_CLK. The divide ratio depends on the clock range setting in the GMAC register 4. The MDC clock is set as shown in Table 3-3 based on the SYS_CLK frequency:

Table 3-3 MDC Generation

CR[3:0] of GMAC Register 4	SYS_CLK	MDC Clock
0000	60 MHz - 100 MHz	SYS_CLK/42
0001	100 MHz - 150 MHz	SYS_CLK/62
0010	20 MHz - 35 MHz	SYS_CLK/16
0011	35 MHz - 60 MHz	SYS_CLK/26
0100	150 MHz - 250 MHz	SYS_CLK/102
0101	250 MHz - 300 MHz	SYS_CLK/124
0110,0111	Reserved	

I/O data of the management operation is input/output to the MDIO signal. The frame structure on the MDIO is shown below.

Figure 3-7 Frame Structure on the MDIO



IDLE: Hi-Z state; there is no clock on MDC

PREAMBLE: 32 continuous bits of value 1

START: Start-of-frame is 01

OPCODE: 10 for Read and 01 for Write

PHY ADDR: 5-bit address select for one of 32 PHYs

REG ADDR: Register address in the selected PHY

TA: When writing, Ethernet-MAC drives 10.
When reading, the 1st bit of the Hi-Z state is driven, and 2nd bit of PHY drives 0.

DATA: When writing, Ethernet-MAC drives write data.
When reading, PHY drives read data.

Management Write Operation

When the user sets the GMII Write and Busy bits (see "4.4. GMAC Register 4 (GAR)(GMII/MII Address register)"), the SMA module transfers the PHY address, the register address in PHY, and the write data to the SMA to initiate a Write operation into the PHY registers. The application should not change the GMII Address register contents or the GMII Data register while the transaction is ongoing.
After the Write operation has completed, resets the Busy bit. The SMA module divides the clock divide ratio that programs

SYS_CLK (CR bits of GMII Address register) to generate the MDC clock for this interface. The GMAC drives the MDIO line for the complete duration of the frame. The frame format for the Write operation is as shown in Figure 3-8:

Figure 3-8 Frame Format at Write

IDLE	PREAMBLE	START	OPCODE	PHY ADDR	REG ADDR	TA	DATA	IDLE
Z	1111…11	01	01	AAAAA	RRRRR	01	DDD…DDD	Z

Management Read Operation

When the user sets the GMII Busy bit (see "4.4. GMAC Register 4 (GAR)(GMII/MII Address register)") with the GMII Write bit as 0, the SMA module transfers the PHY address and the register address in PHY to the SMA to initiate a Read operation in the PHY registers. The application should not change the GMII Address register contents or the GMII Data register while the transaction is ongoing.

After the Read operation has completed, resets the Busy bit and updates the GMII Data register with the data read from the PHY. The frame format for the Read operation is as shown in Figure 3-9:

Figure 3-9 Frame Format at Read

IDLE	PREAMBLE	START	OPCODE	PHY ADDR	REG ADDR	TA	DATA	IDLE
Z	1111…11	01	10	AAAAA	RRRRR	Z0	DDD…DDD	Z

3.8 IEEE 1588

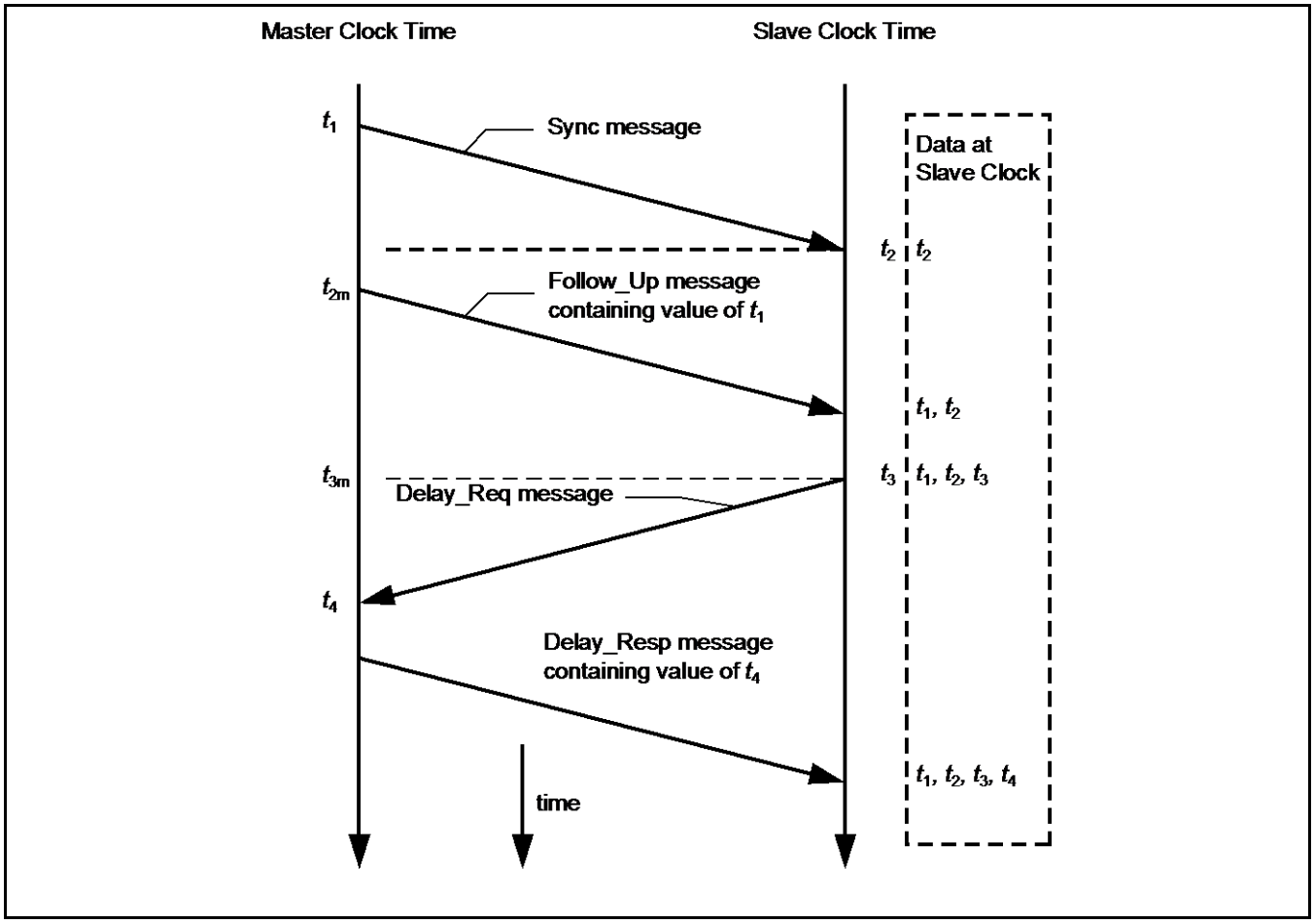
This section explains the functions of IEEE1588.

Overview of IEEE 1588

The IEEE 1588 standard defines a protocol enabling precise synchronization of clocks in measurement and control systems implemented with technologies such as network communication, local computing, and distributed objects. The protocol applies to systems communicating by local area networks supporting multicast messaging, including (but not limited to) Ethernet. This protocol enables heterogeneous systems that include clocks of varying inherent precision, resolution, and stability to synchronize. The protocol supports system-wide synchronization accuracy in the sub-microsecond range with minimal network and local clock computing resources.

The messaged-based protocol, named Precision Time Protocol (PTP), is transported over UDP/IP. The system or network is classified into Master and Slave nodes for distributing the timing/clock information. The protocol's technique for synchronizing a slave node to a master node by exchanging PTP messages is depicted in Figure 3-10.

Figure 3-10 Networked Time Synchronization



1. The master broadcasts PTP Sync messages to all its nodes. The Sync message contains the master's reference time information. The time at which this message leaves the master's system is t1 and must, for Ethernet ports, be captured at the PHY interface.
2. A slave receives the Sync message and also captures the exact time, t2, using its timing reference.
3. The master then sends the slave a Follow_up message, which contains t1 information for later use.

4. The slave sends the master a Delay_Req message, noting the exact time, t3, at which this frame leaves the MII.
5. The master receives this message, capturing the exact time, t4, at which it enters its system.
6. The master sends the t4 information to the slave in the Delay_Resp message.
7. The slave uses the four values of t1, t2, t3, and t4 to synchronize its local timing reference to the master's timing reference.

Most of the protocol implementation occurs in the software, above the UDP layer. As described above, however, hardware support is required to capture the exact time when specific PTP packets enter or leave the Ethernet port. This timing information must be captured and returned to the software for the proper implementation of PTP with high accuracy.

Reference Timing Source

The reference clock input and uses it to generate the Reference time (also called the System Time) internally and use it to capture time stamps. The generation, update, and modification of the System Time are described in System Time Register Module.

To get a snapshot of the time, the Ethernet-MAC requires a reference time in 64-bit format (split into two 32-bit channels, with the upper 32 bits providing time in seconds, and the lower 32 bits indicating time in nanoseconds).

The IEEE 1588-2008 standard defines the seconds field of the time to be 48-bits wide. The fields to time-stamp will be the following.

- UInteger48- seconds field
- UInteger32-nanoseconds field

The seconds field is the integer portion of the timestamp in units of seconds. The nanoseconds field is the fractional portion of the timestamp in units of nanoseconds. E.g. 2.000000001 seconds is represented as secondsField = 0x0000_0000_0002 and nanoSeconds = 0x0000_0001. Thus the maximum value in nanoseconds field in this format will be 0x3B9A_C9FF value (i.e (10e9-1) nanoseconds). This is defined as digital rollover mode of operation. It will also support the older mode in which the nano-seconds field will roll-over and increment the seconds field after the value of 0x7FFF_FFFF. (Accuracy is ~0.466 ns per bit). This is defined as the binary rollover mode. The modes can be controlled using the "TSDB: Time Stamp Digital or Binary rollover control" bit of the time stamp control register.

When the Advanced IEEE 1588 time-stamp feature is selected time maintained in the Ethernet-MAC will still be 64-bit wide, as the overflow to the upper 16-bits of seconds register happens once in 130 years. The value of the upper 16-bits of the seconds field can be obtained only from the GMAC register. A CSR status bit which indicates if the 32-bit "seconds" field has overflowed is also provided.

There is also a pulse-per-second output given to indicate 1 second interval (default). Option is provided to change the interval. See 4.30. GMAC Register 459 (PPSCR) for more information.

Transmit Path Functions

When a frame's SFD(Start Frame Delimiter) is output on the PHY interface, a time stamp is captured. Frames for which capturing a time stamp is required are controllable on a per-frame basis. In other words, each transmit frame can be marked to indicate whether or not a time stamp must be captured for that frame.

No snooping or processing of the transmitted frames is performed to identify PTP frames. Framewise control is exercised either through control bits in the transmit descriptor.

Captured time stamps are returned to the application in a manner similar to that in which status is provided for frames. The time stamp, along with the Transmit status of the frame, is given on this bus. The time stamp is returned to software inside the corresponding transmit descriptor, thus connecting the time stamp automatically to the specific PTP frame. The 64-bit time stamp information is written back to the TDES6 and TDES7 fields, with TDES6 holding the time stamp's 32 least significant bits.

Receive Path Functions

The Ethernet-MAC returns the time-stamp to the software in the corresponding receive descriptor. The 64-bit time stamp information is written back to the RDES6 and RDES7 fields, with RDES6 holding the time stamp's 32 least significant bits. The time stamp is written only to the receive descriptor for which the Last Descriptor status field has been set to 1 (the EOF marker). When the time stamp is not available (for example, due to an receive FIFO overflow) an all-ones pattern is written to the descriptors (RDES6 and RDES7), indicating that time stamp is not correct. If the software uses a control register bit to disable time stamping, the DMA does not alter RDES6 or RDES7.

Time Stamp Error Margin

According to the IEEE 1588 specifications, the time stamp must be captured at the SFD of transmitted and received frames at the PHY interface. Since the reference timing source is different from the PHY interface clocks, a small error margin is introduced, due to the transfer of information across asynchronous clock domains.

In the transmit path, the captured and reported time stamp has a maximum error margin of 2 PTP clocks. In other words, the captured time stamp has the value of the reference time source given within 2 clocks after the SFD has been transmitted on the PHY interface. Similarly, on the receive path, the error margin is 3 PHY interface clocks, plus up to 2 PTP clocks. You can ignore the error margin due to the 3 PHY interface clocks by assuming that this constant delay is present in the system (or link) before the SFD data reaches the PHY interface.

Frequency Range of Reference Timing Clock

Because asynchronous logic is in place for time stamp information transfers across clock domain, a minimum delay is required between two consecutive time stamp captures. This delay is 3 clock cycles of both the PHY interface and PTP clocks. If the gap is shorter, the Ethernet-MAC does not take a time stamp snapshot for the second frame.

The maximum PTP clock frequency is limited by the maximum resolution of the reference time (1 ns resulting in 1 GHz) and the timing constraints achievable for logic operating on the PTP clock. Another factor to consider is that the resolution, or granularity, of the reference time source determines the accuracy of the synchronization. Hence, a higher PTP clock frequency gives better system performance. The minimum PTP clock frequency depends on the time required between two consecutive SFD bytes. Because the PHY interface clock frequency is fixed by IEEE specification, the minimum PTP clock frequency required for proper operation depends on the Ethernet-MAC operating mode and operating speed.

For example, in 100-Mbps full-duplex operation, the minimum gap between two SFDs is 160 PHY interface clocks (128 clocks for a 64-byte frame + 24 clocks of min IFG + 8 clocks of preamble).

In the first example, (3 × PTP) + (3 ×PHY) ≤ 160 ×PHY; thus, the minimum PTP clock frequency is about 0.5 MHz ((160 - 3) × 40 ns ÷ 3 = 2,093 ns period).

System Time Register Module

64-bit time is maintained in this module and updated using the input reference clock (PTP_CLK (Note: The AHB system bus clock:HCLK is connected to the input reference clock (PTP_CLK) in Ethernet-MAC for FM3 series.). This time is the source for taking snapshots (time stamps) of Ethernet frames being transmitted or received.

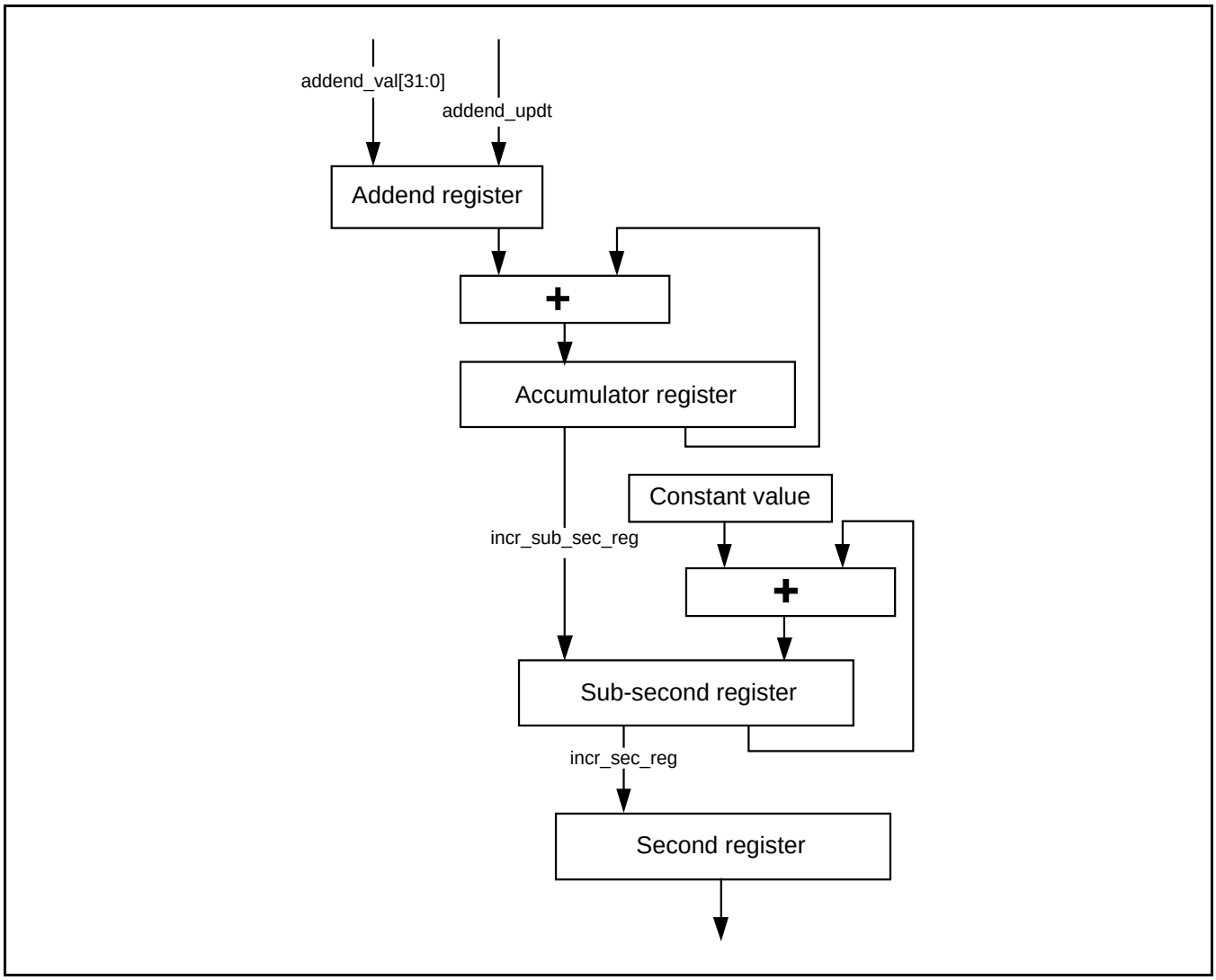
The System Time counter can be initialized or corrected using the course correction method. In this method, the initial value or the offset value is written to the Time Stamp Update register. For initialization, the System Time counter is written with the value in the Time Stamp Update registers, while for system time correction, the offset value is added to or subtracted from the system time.

In the fine correction method, a slave clock's (PTP_CLK) frequency drift with respect to the master clock (as defined in IEEE 1588) is corrected over a period of time instead of in one clock, as in course correction. This helps maintain linear time and does not introduce drastic changes (or a large jitter) in the reference time between PTP Sync message intervals. In this method, an accumulator sums up the contents of the Addend register, as shown in Figure 3-11. The arithmetic carry that the

accumulator generates is used as a pulse to increment the system time counter. The accumulator and the addend are 32-bit registers. Here, the accumulator acts as a high-precision frequency multiplier or divider.

This algorithm is depicted in Figure 3-11:

Figure 3-11 System Time Update Using Fine correction Method



The System Time Update logic requires a 50 MHz clock frequency to achieve 20 ns accuracy. The frequency division is the ratio of the reference clock frequency to the required clock frequency. Hence, if the reference clock (PTP_CLK) is, for example, 66 MHz, this ratio is calculated as 66 MHz / 50 MHz = 1.32. Hence, the default added value to be set in the register is 232 / 1.32, 0xC1F07C1F.

If the reference clock drifts lower, to 65 MHz for example, the ratio is 65 / 50, or 1.3 and the value to set in the addend register is 2^{32} / 1.30, or 0xC4EC4EC4. If the clock drifts higher, to 67 MHz for example, the addend register must be set to 0xBF0B7672. When the clock drift is nil, the default added value of 0xC1F07C1F (2^{32} / 1.32) must be programmed.

The software must calculate the drift in frequency based on the Sync messages and update the Addend register accordingly. Initially, the slave clock is set with FreqCompensationValue0 in the Addend register. This value is as follows:

FreqCompensationValue0 = 2^{32} / FreqDivisionRatio

If MasterToSlaveDelay is initially assumed to be the same for consecutive Sync messages, the algorithm described below must be applied. After a few Sync cycles, frequency lock occurs. The slave clock can then determine a precise MasterToSlaveDelay value and re-synchronize with the master using the new value.

The algorithm is as follows:

- At time MasterSyncTimen the master sends the slave clock a Sync message. The slave receives this message when its local clock is SlaveClockTimen and computes MasterClockTimen as:
  MasterClockTimen = MasterSyncTimen + MasterToSlaveDelayn
- The master clock counts for current Sync cycle, MasterClockCountn is given by:
  MasterClockCountn = MasterClockTimen - MasterClockTimenn-1 (assuming that MasterToSlaveDelay is the same for Sync cycles n and n - 1)
- The slave clock counts for current Sync cycle, SlaveClockCountn is given by:
  SlaveClockCountn = SlaveClockTimen - SlaveClockTimen - 1
- The difference between master and slave clock counts for current Sync cycle, ClockDiffCountn is given by:
  ClockDiffCountn = MasterClockCountn - SlaveClockCountn
- The frequency-scaling factor for slave clock, FreqScaleFactorn is given by:
  FreqScaleFactorn = (MasterClockCountn + ClockDiffCountn) / SlaveClockCountn
- The frequency compensation value for Addend register, FreqCompensationValuen is given by:
  FreqCompensationValuen = FreqScaleFactorn × FreqCompensationValuen - 1

In theory, this algorithm achieves lock in one Sync cycle; however, it may take several cycles, due to changing network propagation delays and operating conditions.
This algorithm is self-correcting: if for any reason the slave clock is initially set to a value from the master that is incorrect, the algorithm will correct it at the cost of more Sync cycles.

PTP Processing and Control

The common message header for PTP messages is shown below. This format conforms to IEEE standard 1588-2008.

Table 3-4 Messages Format Defined in IEEE 1588-2008

BITS		OCTETS	OFFSET
transportSpecific	messageType	1	0
Reserved	versionPTP	1	1
messageLength		2	2
domainNumber		1	4
Reserved		1	5
flagField		2	6
correctionField		8	8
Reserved		4	16
sourcePortIdentity		10	20
sequenceId		2	30
messageType field		1	32
logMessageInterva		1	33

There are some fields in the PTP frame that are used to detect the type and control the snapshot to be generated. This is different for PTP frames sent directly over Ethernet, PTP frames sent over UDP / IPv4 and PTP frames that are sent over UDP / IPv6. The following sections provide information on the fields that are used to control the generation of the snapshot.

PTP Frame over IPv4

Table 3-5 provides information about the fields that are matched to control snapshot for the PTP packets sent over UDP over IPv4 for IEEE 1588 version 1 and 2. The octet positions for the tagged frames are offset by 4. This is based on Annex D of IEEE 1588-2008 standard and the message format defined in Table 3-4.

Table 3-5 IPv4-UDP PTP Frame Fields Required for Control and Status

Field Matched	Octet Position	Matched Value	Description
MAC Frame Type	12,13	0x0800	IPv4 datagram
IP Version and Header Length	14	0x45	IP version is IPv4
Layer 4 protocol	23	0x11	UDP
IP Multicast address	30,31,32,33	0xE0,0x00, 0x01,0x81(Hex) 0xE0,0x00,0x00,0x6B(Hex)	PTP-primary multicast address allowed 224.0.1.129 224.0.1.130 224.0.1.131 224.0.1.132
UDP destination port	36,37	0x013F,0x0140	0x013F:PTP event message(*) 0x0140:PTP general messages
PTP Control Field (IEEE version 1)	74	0x00/0x01/0x02/0x03/0x04	0x00:SYNC, 0x01:Delay_Req 0x02:Follow_Up 0x03:Delay_Resp 0x04:Management
PTP Message Type Field (IEEE version 2)	42	0x0/0x1/0x2/0x3/0x8/ 0x9/0xA/0xB/0xC/0xD	0x0:SYNC 0x1:Delay_Req 0x2:Pdelay_Req 0x3:Pdelay_Resp 0x8:Follow_Up 0x9:Delay_Resp 0xA:Pdelay_Resp_Follow_Up 0xB:Announce 0xC:Signaling 0xD:Management
PTP version field	43(nibble)	0x1 or 0x2	0x1:Supports PTP version 1 0x2:Supports PTP version 2

(*) PTP event messages are SYNC, Delay_Req (IEEE 1588 version 1 and 2) or Pdelay_Req, Pdelay_Resp (IEEE 1588 version 2 only).

PTP Frame Over IPv6

Table 3-6 provides information about the fields that are matched to control the snapshots for the PTP packets. This information is sent over UDP over IPv6 for IEEE 1588 version 1 and 2. The octet positions for the tagged frames are offset by 4. This is based on Annex D of the IEEE 1588-2008 standard and the message format defined in Table 3-4.

Table 3-6 IPv6-UDP PTP Frame Fields Required for Control and Status

Field Matched	Octet Position	Matched Value	Description
MAC Frame Type	12,13	0x86DD	IPv4 datagram
IP version	14 (bits [7:4])	0x6	IP version is IPv6
Layer 4 protocol	20(*)	0x11	UDP
PTP Multicast address	38-53	FF0x:0:0:0:0:0:0:181 (Hex) FF02:0:0:0:0:0:0:6B (Hex)	PTP- primary multicast address: FF0x:0:0:0:0:0:0:181(Hex) PTP - Pdelay multicast address: FF02:0:0:0:0:0:0:6B(Hex)
UDP destination port	56,57	0x013F,0x0140	0x013F:PTP event message 0x0140:PTP general messages
PTP Control Field (IEEE 1588 version 1)	93 (*)	0x00/0x01/0x02/0x03/0x04	0x00:SYNC 0x01:Delay_Req 0x02:Follow_Up 0x03:Delay_Resp 0x04:Management (version1)
PTP Message Type Field (IEEE version 2)	74(*)(nibble)	0x0/0x1/0x2/0x3/0x8/ 0x9/0xA/0xB/0xC/0xD	0x0:SYNC 0x1:Delay_Req 0x2:Pdelay_Req 0x3:Pdelay_Resp 0x8:Follow_Up 0x9:Delay_Resp 0xA:Pdelay_Resp_Follow_Up 0xB:Announce 0xC:Signaling 0xD:Management
PTP version field	75(nibble)	0x1 or 0x2	0x1:Supports PTP version 1 0x2:Supports PTP version 2

(*) The Extension Header is not defined for PTP packets.

PTP Frame Over Ethernet

Table 3-7 gives the details of the fields that will be matched to control snapshot for PTP packets sent over Ethernet. Note that the octet positions for tagged frames will be offset by 4.

Table 3-7 Ethernet PTP Frame Fields Required for Control and Status

Field Matched	Octet Position	Matched Value	Description
MAC Destination Multicast Address(*a)	0-5	01-1B-19-00-00-00 01-80-C2-00-00-0E	All PTP messages can use any of the following multicast addresses(*b): 01-1B-19-00-00-00 01-80-C2-00-00-0E(*c)
MAC Frame Type	12,13	0x86F7	PTP Ethernet frame
PTP Control Field (IEEE version 1)	45	0x00/0x01/0x02/0x03/0x04	0x00:SYNC 0x01:Delay_Req 0x02:Follow_Up 0x03:Delay_Resp 0x04:Management
PTP Message Type Field (IEEE version 2)	14(nibble)	0x0/0x1/0x2/0x3/0x8/ 0x9/0xA/0xB/0xC/0xD	0x0:SYNC 0x1:Delay_Req 0x2:Pdelay_Req 0x3:Pdelay_Resp 0x8:Follow_Up 0x9:Delay_Resp 0xA:Pdelay_Resp_Follow_Up 0xB:Announce 0xC:Signaling 0xD:Management
PTP version field	15(nibble)	0x1 or 0x2	0x1:Supports PTP version 1 0x2:Supports PTP version 2

*a: The address match of destination addresses (DA) programmed in MAC address 0 to 31 is used if the control bit 18 (TSENMF: Enable MAC address for PTP frame filtering) of the Timestamp Control register is set.

*b: IEEE standard 1588-2008, Annex F

*c: The GMAC-UNIV does not consider the PTP version messages with Peer delay multicast address (01-80-C2-00-00- 0E) as valid PTP messages.

4. Registers

This section explains the register function of Ethernet-MAC.

Register Map

Table 4-1 shows the register list of Ethernet-MAC.

Table 4-1 Register List of Ethernet-MAC

Address	Register No.	Abbreviation	Register Full Name	Ref.
0x0000	GMAC Reg. 0	MCR	MAC Configuration register	4.1
0x0004	GMAC Reg. 1	MFFR	MAC Frame Filter register	4.2
0x0008	GMAC Reg. 2	MHTRH	MAC Hash Table register (High)	4.3
0x000C	GMAC Reg. 3	MHTRL	MAC Hash Table register (Low)	4.3
0x0010	GMAC Reg. 4	GAR	GMII Address register	4.4
0x0014	GMAC Reg. 5	GDR	GMII Data register	4.5
0x0018	GMAC Reg. 6	FCR	Flow Control register	4.6
0x001C	GMAC Reg. 7	VTR	VLAN Tag register	4.7
0x0020 - 0x0024	-	-	Reserved	-
0x0028	GMAC Reg. 10	RWFFR	Remote Wake-up Frame Filter register	4.8
0x002C	GMAC Reg. 11	PMTR	PMT register	4.9
0x0030	GMAC Reg. 12	LPICSR	LPI Control and Status register	4.10
0x0034	GMAC Reg. 13	LPITCR	LPI Timers Control register	4.11
0x0038	GMAC Reg. 14	ISR	Interrupt Status register	4.12
0x003C	GMAC Reg. 15	IMR	Interrupt Mask register	4.13
0x0040	GMAC Reg. 16	MAR0H	MAC Address0 register (High)	4.14
0x0044	GMAC Reg. 17	MAR0L	MAC Address0 register (Low)	4.15
0x0048	GMAC Reg. 18	MAR1H	MAC Address1 register (High)	4.16
0x004C	GMAC Reg. 19	MAR1L	MAC Address1 register (Low)	4.17
0x0050	GMAC Reg. 20	MAR2H	MAC Address2 register (High)	4.16
0x0054	GMAC Reg. 21	MAR2L	MAC Address2 register (Low)	4.17
0x0058	GMAC Reg. 22	MAR3H	MAC Address3 register (High)	4.16
0x005C	GMAC Reg. 23	MAR3L	MAC Address3 register (Low)	4.17
0x0060	GMAC Reg. 24	MAR4H	MAC Address4 register (High)	4.16
0x0064	GMAC Reg. 25	MAR4L	MAC Address4 register (Low)	4.17
0x0068	GMAC Reg. 26	MAR5H	MAC Address5 register (High)	4.16
0x006C	GMAC Reg. 27	MAR5L	MAC Address5 register (Low)	4.17
0x0070	GMAC Reg. 28	MAR6H	MAC Address6 register (High)	4.16
0x0074	GMAC Reg. 29	MAR6L	MAC Address6 register (Low)	4.17
0x0078	GMAC Reg. 30	MAR7H	MAC Address7 register (High)	4.16
0x007C	GMAC Reg. 31	MAR7L	MAC Address7 register (Low)	4.17
0x0080	GMAC Reg. 32	MAR8H	MAC Address8 register (High)	4.16
0x0084	GMAC Reg. 33	MAR8L	MAC Address8 register (Low)	4.17
0x0088	GMAC Reg. 34	MAR9H	MAC Address9 register (High)	4.16
0x008C	GMAC Reg. 35	MAR9L	MAC Address9 register (Low)	4.17
0x0090	GMAC Reg. 36	MAR10H	MAC Address10 register (High)	4.16
0x0094	GMAC Reg. 37	MAR10L	MAC Address10 register (Low)	4.17

Address	Register No.	Abbreviation	Register Full Name	Ref.
0x0098	GMAC Reg. 38	MAR11H	MAC Address11 register (High)	4.16
0x009C	GMAC Reg. 39	MAR11L	MAC Address11 register (Low)	4.17
0x00A0	GMAC Reg. 40	MAR12H	MAC Address12 register (High)	4.16
0x00A4	GMAC Reg. 41	MAR12L	MAC Address12 register (Low)	4.17
0x00A8	GMAC Reg. 42	MAR13H	MAC Address13 register (High)	4.16
0x00AC	GMAC Reg. 43	MAR13L	MAC Address13 register (Low)	4.17
0x00B0	GMAC Reg. 44	MAR14H	MAC Address14 register (High)	4.16
0x00B4	GMAC Reg. 45	MAR14L	MAC Address14 register (Low)	4.17
0x00B8	GMAC Reg. 46	MAR15H	MAC Address15 register (High)	4.16
0x00BC	GMAC Reg. 47	MAR15L	MAC Address15 register (Low)	4.17
0x00C0 - 0x00D0	-	-	Reserved	-
0x00D8	GMAC Reg. 54	RGSR	RGMII Status register	4.18
0x00DC - 0x00FC	-	-	Reserved	-
0x0100	GMAC Reg. 64	mmc_cntl	MMC Control register	4.49
0x0104	GMAC Reg. 65	mmc_intr_rx	MMC Receive Interrupt register	4.50
0x0108	GMAC Reg. 66	mmc_intr_tx	MMC Transmit Interrupt register	4.51
0x010C	GMAC Reg. 67	mmc_intr_mask_rx	MMC Receive Interrupt Mask register	4.52
0x0110	GMAC Reg. 68	mmc_intr_mask_tx	MMC Transmit Interrupt Mask register	4.53
0x0114	GMAC Reg. 69	txoctetcount_gb	MMC Counters	4.48
0x0118	GMAC Reg. 70	txframecount_gb		
0x011C	GMAC Reg. 71	txbroadcastframes_g		
0x0120	GMAC Reg. 72	txmulticastframes_g		
0x0124	GMAC Reg. 73	tx64octets_gb		
0x0128	GMAC Reg. 74	tx65to127octets_gb		
0x012C	GMAC Reg. 75	tx128to255octets_gb		
0x0130	GMAC Reg. 76	tx256to511octets_gb		
0x0134	GMAC Reg. 77	t x512to1023octets_gb		
0x0138	GMAC Reg. 78	tx1024tomaxoctets_gb		
0x013C	GMAC Reg. 79	txunicastframes_gb		
0x0140	GMAC Reg. 80	txmulticastframes_gb		
0x0144	GMAC Reg. 81	txbroadcastframes_gb		
0x0148	GMAC Reg. 82	txunderflowerror		
0x014C	GMAC Reg. 83	txsinglecol_g		
0x0150	GMAC Reg. 84	txmulticol_g		
0x0154	GMAC Reg. 85	txdeferred		
0x0158	GMAC Reg. 86	txlatecol		
0x015C	GMAC Reg. 87	txexesscol		
0x0160	GMAC Reg. 88	txcarriererrror		
0x0164	GMAC Reg. 89	txoctetcount_g		
0x0168	GMAC Reg. 90	txframecount_g		
0x016C	GMAC Reg. 91	txexecessdef	MMC Counters	4.48
0x0170	GMAC Reg. 92	txpauseframes		
0x0174	GMAC Reg. 93	txvlanframes_g		

Address	Register No.	Abbreviation	Register Full Name	Ref.
0x0178, 0x017C	-	-	Reserved	-
0x0180	GMAC Reg. 96	rxframecount_gb	MMC Counters	4.48
0x0184	GMAC Reg. 97	rxoctetcount_gb		
0x0188	GMAC Reg. 98	rxoctetcount_g		
0x018C	GMAC Reg. 99	rxbroadcastframes_g		
0x0190	GMAC Reg. 100	rxmulticastframes_g		
0x0194	GMAC Reg. 101	rxcrcerror		
0x0198	GMAC Reg. 102	rxallignmenterror		
0x019C	GMAC Reg. 103	rxrunterror		
0x01A0	GMAC Reg. 104	rxjabbererror		
0x01A4	GMAC Reg. 105	rxundersize_g		
0x01A8	GMAC Reg. 106	rxoversize_g		
0x01AC	GMAC Reg. 107	rx64octets_gb		
0x01B0	GMAC Reg. 108	rx65to127octets_gb		
0x01B4	GMAC Reg. 109	rx128to255octets_gb		
0x01B8	GMAC Reg. 110	rx256to511octets_gb		
0x01BC	GMAC Reg. 111	rx512to1023octets_gb		
0x01C0	GMAC Reg. 112	rx1024tomaxoctets_gb		
0x01C4	GMAC Reg. 113	rxunicastframes_g		
0x01C8	GMAC Reg. 114	rxlengtherror		
0x01CC	GMAC Reg. 115	rxoutofrangetype		
0x01D0	GMAC Reg. 116	rxpauseframes		
0x01D4	GMAC Reg. 117	rxfifooverflow	MMC Counters	4.48
0x01D8	GMAC Reg. 118	rxvlanframes_gb		
0x01DC	GMAC Reg. 119	rxwatchdogerror		
0x01E0 - 0x01FC	-	-	Reserved	-
0x0200	GMAC Reg. 128	mmc_ipc_intr_mask_rx	MMC Receive Checksum Offload Interrupt Mask register	4.54
0x0204	-	-	Reserved	-
0x0208	GMAC Reg. 130	mmc_ipc_intr_rx	MMC Receive Checksum Offload Interrupt register	4.55
0x020C	-	-	Reserved	-
0x0210	GMAC Reg. 132	rxipv4_gd_frms	MMC Counters	4.48
0x0214	GMAC Reg. 133	rxipv4_hdrerr_frms		
0x0218	GMAC Reg. 134	rxipv4_nopay_frms		
0x021C	GMAC Reg. 135	rxipv4_frag_frms		
0x0220	GMAC Reg. 136	rxipv4_udsbl_frms		
0x0224	GMAC Reg. 137	rxipv6_gd_frms		
0x0228	GMAC Reg. 138	rxipv6_hdrerr_frms		
0x022C	GMAC Reg. 139	rxipv6_nopay_frms		
0x0230	GMAC Reg. 140	rxudp_gd_frms		
0x0234	GMAC Reg. 141	rxudp_err_frms		
0x0238	GMAC Reg. 142	rxtcp_gd_frms		
0x023C	GMAC Reg. 143	rxtcp_err_frms		

Address	Register No.	Abbreviation	Register Full Name	Ref.
0x0240	GMAC Reg. 144	rxicmp_gd_frms	MMC Counters	4.48
0x0244	GMAC Reg. 145	rxicmp_err_frms		
0x0248, 0x024C	-	-	Reserved	-
0x0250	GMAC Reg. 148	rxipv4_gd_octets	MMC Counters	4.48
0x0254	GMAC Reg. 149	rxipv4_hdrerr_octets		
0x0258	GMAC Reg. 150	rxipv4_nopay_octets	MMC Counters	4.48
0x025C	GMAC Reg. 151	rxipv4_frag_octets		
0x0260	GMAC Reg. 152	rxipv4_udsbl_octets		
0x0264	GMAC Reg. 153	rxipv6_gd_octets		
0x0268	GMAC Reg. 154	rxipv6_hdrerr_octets		
0x026C	GMAC Reg. 155	rxipv6_nopay_octets		
0x0270	GMAC Reg. 156	rxudp_gd_octets		
0x0274	GMAC Reg. 157	rxudp_err_octets		
0x0278	GMAC Reg. 158	rxtcp_gd_octets		
0x027C	GMAC Reg. 159	rxtcp_err_octets		
0x0280	GMAC Reg. 160	rxicmp_gd_octets		
0x0284	GMAC Reg. 161	rxicmp_err_octets		
0x0288, 0x06FC	-	-	Reserved	-
0x0700	GMAC Reg. 448	TSCR	Time Stamp Control register	4.19
0x0704	GMAC Reg. 449	SSIR	Sub-Second Increment register	4.20
0x0708	GMAC Reg. 450	STSR	System Time - Seconds register	4.21
0x070C	GMAC Reg. 451	STNR	System Time - Nanoseconds register	4.22
0x0710	GMAC Reg. 452	STSUR	System Time - Seconds Update register	4.23
0x0714	GMAC Reg. 453	STSNUR	System Time - Nanoseconds Update register	4.24
0x0718	GMAC Reg. 454	TSAR	Time Stamp Addend register	4.25
0x071C	GMAC Reg. 455	TTSR	Target Time Seconds register	4.26
0x0720	GMAC Reg. 456	TTNR	Target Time Nanoseconds register	4.27
0x0724	GMAC Reg. 457	STHWSR	System Time - High Word Seconds register	4.28
0x0728	GMAC Reg. 458	TSR	Time Stamp Status register	4.29
0x072C	GMAC Reg. 459	PPSCR	PPC Control register	4.30
0x0730	GMAC Reg. 460	ATNR	Auxiliary Time Stamp-Nanosecond register	4.31
0x0734	GMAC Reg. 461	ATSR	Auxiliary Time Stamp-Seconds register	4.32
0x0738 - 0x07FC	-	-	Reserved	-
0x0800	GMAC Reg. 512	MAR16H	MAC Address16 register (High)	4.16
0x0804	GMAC Reg. 513	MAR16L	MAC Address16 register (Low)	4.17
0x0808	GMAC Reg. 514	MAR17H	MAC Address17 register (High)	4.16
0x080C	GMAC Reg. 515	MAR17L	MAC Address17 register (Low)	4.17
0x0810	GMAC Reg. 516	MAR18H	MAC Address18 register (High)	4.16
0x0814	GMAC Reg. 517	MAR18L	MAC Address18 register (Low)	4.17
0x0818	GMAC Reg. 518	MAR19H	MAC Address19 register (High)	4.16
0x081C	GMAC Reg. 519	MAR19L	MAC Address19 register (Low)	4.17
0x0820	GMAC Reg. 520	MAR20H	MAC Address20 register (High)	4.16

Address	Register No.	Abbreviation	Register Full Name	Ref.
0x0824	GMAC Reg. 521	MAR20L	MAC Address20 register (Low)	4.17
0x0828	GMAC Reg. 522	MAR21H	MAC Address21 register (High)	4.16
0x082C	GMAC Reg. 523	MAR21L	MAC Address21 register (Low)	4.17
0x0830	GMAC Reg. 524	MAR22H	MAC Address22 register (High)	4.16
0x0834	GMAC Reg. 525	MAR22L	MAC Address22 register (Low)	4.17
0x0838	GMAC Reg. 526	MAR23H	MAC Address23 register (High)	4.16
0x083C	GMAC Reg. 527	MAR23L	MAC Address23 register (Low)	4.17
0x0840	GMAC Reg. 528	MAR24H	MAC Address24 register (High)	4.16
0x0844	GMAC Reg. 529	MAR24L	MAC Address24 register (Low)	4.17
0x0848	GMAC Reg. 530	MAR25H	MAC Address25 register (High)	4.16
0x084C	GMAC Reg. 531	MAR25L	MAC Address25 register (Low)	4.17
0x0850	GMAC Reg. 532	MAR26H	MAC Address26 register (High)	4.16
0x0854	GMAC Reg. 533	MAR26L	MAC Address26 register (Low)	4.17
0x0858	GMAC Reg. 534	MAR27H	MAC Address27 register (High)	4.16
0x085C	GMAC Reg. 535	MAR27L	MAC Address27 register (Low)	4.17
0x0860	GMAC Reg. 536	MAR28H	MAC Address28 register (High)	4.16
0x0864	GMAC Reg. 537	MAR28L	MAC Address28 register (Low)	4.17
0x0868	GMAC Reg. 538	MAR29H	MAC Address29 register (High)	4.16
0x086C	GMAC Reg. 539	MAR29L	MAC Address29 register (Low)	4.17
0x0870	GMAC Reg. 540	MAR30H	MAC Address30 register (High)	4.16
0x0874	GMAC Reg. 541	MAR30L	MAC Address30 register (Low)	4.17
0x0878	GMAC Reg. 542	MAR31H	MAC Address31 register (High)	4.16
0x087C	GMAC Reg. 543	MAR31L	MAC Address31 register (Low)	4.17
0x0880 - 0x0FFC	-	-	Reserved	-
0x1000	DMA Reg. 0	BMR	BUS Mode register	4.33
0x1004	DMA Reg. 1	TPDR	Transmit Poll Demand register	4.34
0x1008	DMA Reg. 2	RPDR	Receive Poll Demand register	4.35
0x100C	DMA Reg. 3	RDLAR	Receive Descriptor List Address register	4.36
0x1010	DMA Reg. 4	TDLAR	Transmit Descriptor List Address register	4.37
0x1014	DMA Reg. 5	SR	Status register	4.38
0x1018	DMA Reg. 6	OMR	Operation Mode register	4.39
0x101C	DMA Reg. 7	IER	Interrupt Enable register	4.40
0x1020	DMA Reg. 8	MFBOCR	Missed Frame and Buffer Overflow Counter register	4.41
0x1024	DMA Reg. 9	RIWTR	Receive Interrupt Watchdog Timer register	4.42
0x1028	-	-	Reserved	-
0x102C	DMA Reg. 11	AHBSR	AHB Status register	4.43
0x1030-1044	-	-	Reserved	-
0x1048	DMA Reg. 18	CHTDR	Current Host Transmit Descriptor register	4.44
0x104C	DMA Reg. 19	CHRDR	Current Host Receive Descriptor register	4.45
0x1050	DMA Reg. 20	CHTBAR	Current Host Transmit Buffer Address register	4.46
0x1054	DMA Reg. 21	CHRBAR	Current Host Receive Buffer Address register	4.47
0x1058 - 0xFFFF	-	-	Reserved	-

How to use Register Bit Maps (Example)

Register Abbreviation (Register full name) Offset address xxxxh

bit	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
Field	A	B	C	D	E	F	G	H
Attribute	R/W	R/W	R	R	R	R	R	R/WSC
Initial value	0	0	0	0	0	0	0	0

Register Abbreviation: Abbreviation of register

Register full name: Full name of register

Offset address: Register offset address

bit: Bit position

Field: Bit Field name

Attribute: Read/write attribute of register. Meaning of the described symbols is as follows.

R: Read enabled

W: Write enabled

R/W_SC: Register field can be read and written by the application (Read and Write), and is cleared to 0 by the GMAC/DMA (Self Clear). The conditions under which the GMAC/DMA clears this field are explained in detail in the field's description.

R_SS_WC: Register field can be read by the application (Read), can be set to 1 by the GMAC/DMA on a certain internal event (Self Set), and can be cleared to 0 by the application with a register write of 1 (Write Clear). A register write of 0 by the application has no effect on this field. The conditions under which the GMAC/DMA sets this field are explained in detail in the field's description. (Example, interrupt bits.)

R/WS_SC: Register field can be read by the application (Read), can be set to 1 by the application. A register write of 0 by the application is invalid. The bit is cleared to 0 by the GMAC/DMA (Self Clear). The conditions under which the GMAC/DMA clears this field are explained in detail in the field's description. (Example, reset signals)

R/SS_SC_WC: Register field can be read by the application (Read), can be set to 1 by the GMAC/DMA on a certain internal event (Self Set), and can be cleared to 0 (Self Clear). The bit can be cleared to 0 by the application with a register write of 0 (Write Clear). A register write of 1 to this bit by the application has no effect on this field. The conditions under which the GMAC/DMA sets or clears this field are explained in detail in the field's description.

R/WT: Register field can be read by the application (Read), and when a write operation is performed with any data value (Write Trigger), an event is triggered, as explained in the field's description. (Example: Receive Poll Demand register)

R_SS_RC: Register field can be read by the application (Read), can be set to 1 by the GMAC/DMA on a certain internal event (Self Set), and is automatically cleared to 0 on a register read (Read Clear). A register write of 0 by the application has no effect on this field. The conditions under which the GMAC/DMA sets this field are explained in detail in the field's description. (Example: Overflow counter.)

R/WSU: Register field can be read and written by the application (Read and Write). The register field updates itself based on the event (Self Update). For example, use for system time in PTP configuration.

Reserved: Register field should not be changed. If a Reserved field is changed, GMAC/DMA runs unexpected operation.

Initial value: Value immediately after resetting of the register.

4.1 GMAC Register 0 (MCR)

MCR register establishes receive and transmit operating modes.

MCR (MAC Configuration register) Address 0000h

bit	31	30	29	28	27	26	25	24
Field	Reserved						CST	TC
Attribute	R	R	R	R	R	R	R/W	R/W
Initial value	-	-	-	-	-	-	0	0

bit	23	22	21	20	19	18	17	16
Field	WD	JD	BE	JE	IFG[2:0]			DCRS
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

bit	15	14	13	12	11	10	9	8
Field	PS	FES	DO	LM	DM	IPC	DR	LUD
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	ACS	BL[1:0]		DC	TE	RE	Reserved	
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R	R
Initial value	0	0	0	0	0	0	-	-

[bit25] CST (CRC stripping for Type frames)

When this bit is set to 1, the last 4 bytes (FCS) of all frames of Ether type (type field greater than 0x0600) will be stripped and dropped before forwarding the frame to the application.

[bit24] TC (Transmit Configuration in RGMII)

This bit is the reserved bit in MII/RMII mode. Always write 0 to this bit.

[bit23] WD (Watchdog Disable)

When this bit is set to 1, the GMAC disables the watchdog timer on the receiver, and can receive frames of up to 16,384 bytes. When this bit is reset, the GMAC allows no more than 2,048 bytes (10,240 if the JE bit is set to 1) of the frame being received and cuts off any bytes received after that.

[bit22] JD (Jabber Disable)

When this bit is set to 1, the GMAC disables the jabber timer on the transmitter, and can transmit frames of up to 16,384 bytes. When this bit is reset, the GMAC cuts off the transmitter if the application sends out more than 2,048 bytes of data (10,240 if the JE bit is set to 1) during transmission.

[bit21] BE (Frame Burst Enable)

This bit is reserved in MII/RMII mode. Always write "0" to this bit.

[bit20] JE (Jumbo Frame Enable)

When this bit is set to 1, GMAC allows to receive Jumbo frames of 9,018 bytes (9,022 bytes for VLAN tagged frames).

[bit19:17] IFG (Inter-Frame GAP)

These bits control the minimum IFG between frames during transmission.

000 : 96 bit times
001 : 88 bit times
010 : 80 bit times
011 : 72 bit times
100 : 64 bit times
101 : 56 bit times
110 : 48 bit times
111 : 40 bit times

Note that in Half-Duplex mode, the minimum IFG can be configured for 64-bit times (IFG = 100) only. Value less than 64-bit times are not considered.

[bit16] DCRS (Disable Carrier Sense During Transaction)

When this bit is set to 1, this bit makes the GMAC transmitter ignore the CRS signal during frame transmission in Half-Duplex mode. This request results in no errors generated due to Loss of Carrier or No Carrier during such transmission. When this bit is 0, the GMAC transmitter generates such errors due to Carrier Sense and will even abort the transmissions.

[bit15] PS (Port Select)

The initial value of this bit is 0. This bit must be set to "1" when initializing GMAC. MII or RMII is selected for the PHY interface.
Always write "1" to this bit.

[bit14] FES (Speed)

Specifies the communication speed in RMII mode:

0: 10 Mbps
1: 100 Mbps

This bit is enabled only in RMII mode.

[bit13] DO (Disable Receive Own)

When this bit is set to "1", the GMAC disables the reception of frames when the TX_EN is asserted in Half-Duplex mode. When this bit is reset to "0", the GMAC receives all packets that are given by the PHY while transmitting. This bit is not applicable if the GMAC is operating in Full-Duplex mode.

[bit12] LM (Loop-back Mode)

When this bit is set to "1", GMAC operates in loop-back mode.
Receive clock input (RX_CLK/REF_CLK) is required for the loopback to work properly.
*When this bit is set, PAUSE Frame can’t be transmitted.

[bit11] DM (Duplex mode)

When this bit is set to 1, GMAC operates in a Full-Duplex mode where it can transmit and receive simultaneously.

[bit10] IPC (Checksum Offload)

When this bit set to 1, enables IPv4 checksum checking for received frame and checking of payloads’ TCP/UDP/ICMP headers. When this bit is reset to 0, the COE function in the receiver is disabled and the corresponding PCE and IP HCE status bits are always cleared.

Note:

- *In Ethernet environment using IPv6 Authentication Header,IPC must be reset to 0.*

[bit9] DR (Disable Retry)

When this bit is set to "1", the GMAC will attempt only one transmission. When a collision occurs on the PHY interface, the GMAC will ignore the current frame transmission and report a Frame Abort and excessive collision error in the transmit frame status. When this bit is reset to "0", the GMAC will attempt retries based on the settings of BL. This bit is applicable only to Half-Duplex mode.

[bit8] LUD (Link Up/Down in RGMII)

This bit is reserved in MII/RMII mode. Always write "0" to this bit.

[bit7] ACS (Automatic Pad/CRC Stripping)

When this bit is set, the GMAC strips the Pad/FCS field on incoming frames only received frames with if the length's field value is less than or equal to 1,500 bytes. All received frames with length field greater than or equal to 1,501 bytes are passed to the application without stripping the Pad/FCS field. When this bit is reset, the GMAC will pass all incoming frames to the Host unmodified.

Note:

- *ACS can be set only in the Receive-Store-Forward mode (setting DMA register 6, OMR bit25 RSF).*

[bit6, 5] BL (Back-off Limit)

The Back-off limit determines the random integer number (r) of slot time delays (512 bit times for 10/100 Mbps). GMAC waits before rescheduling a transmission attempt during retries after a collision. This bit is applicable only to Half-Duplex mode.

00 : k = min (n, 10) (default)
01 : k = min (n, 8)
10 : k = min (n, 4)
11 : k = min (n, 1)
where n = retransmission retry attempt. The random integer r takes the value in the range $0 \leq r < 2^k$

[bit4] DC (Deferral Check)

When this bit is set, the deferral check function is enabled in the GMAC. The GMAC will issue a Frame Abort status, along with the excessive deferral error bit set in the transmit frame status when the transmit state machine is deferred for more than 24,288 bit times in 10/100Mbps mode. If the Jumbo frame mode is enabled in 10/100 Mbps mode, the threshold for deferral is 155,680 bit times. Deferral begins when the transmitter is ready to transmit, but is prevented because of an active CRS (carrier sense) signal on the PHY interface. Defer time is not cumulative. If the transmit defers for 10,000 bit times, then transmits, collides, backs off, and then has to defer again after completion of back-off, the deferral timer resets to 0 and restarts. When this bit is reset, the deferral check function is disabled and the GMAC defers until the CRS signal goes inactive. This bit is applicable only in Half-Duplex mode.

[bit3] TE (Transmitter Enable)

When this bit is set to 1, the transmit state machine of the GMAC is enabled for transmission on the PHY interface. When this bit is reset to 0, the GMAC transmit state machine is disabled after the completion of the transmission of the current frame, and will not transmit any further frames.

[bit2] RE (Receiver Enable)

When this bit is set to "1", the receive state machine of the GMAC is enabled for receiving frames from the PHY interface. When this bit is reset to "0", the receive state machine of the GMAC is disabled and will not receive any frames from the PHY interface.

4.2 GMAC Register 1 (MFFR)

The MFFR register contains the filter controls for receiving frames.

Some of the controls from this register go to the address check block of the GMAC, which performs the first level of address filtering. The second level of filtering is performed on the incoming frame, based on other controls such as Pass Bad Frames and Pass Control Frames.

MFFR								Address 00004h
bit	31	30	29	28	27	26	25	24
Field	RA	Reserved						
Attribute	R/W	R	R	R	R	R	R	R
Initial value	0	-	-	-	-	-	-	-
bit	23	22	21	20	19	18	17	16
Field	Reserved							
Attribute	R	R	R	R	R	R	R	R
Initial value	-	-	-	-	-	-	-	-
bit	15	14	13	12	11	10	9	8
Field	Reserved					HPF	SAF	SAIF
Attribute	R	R	R	R	R	R/W	R/W	R/W
Initial value	-	-	-	-	-	-	0	0
bit	7	6	5	4	3	2	1	0
Field	PCF[1: 0]		DB	PM	DAIF	HMC	HUC	PR
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

[bit31] RA (Receive All)

When this bit is set, the GMAC Receiver module passes all frames received to the Application irrespective of whether they pass the address filter. The result of the SA/DA filtering is updated (pass or fail) in the corresponding bits in the Receive Status. When this bit is reset, the Receiver module passes only those frames that pass the SA/DA address filter to the Application.

[bit10] HPF (Hash or Perfect Filter)

When this bit is set to 1, this bit configures the address filter to pass a frame if it matches either the perfect filtering or the hash filtering as set by HMC or HUC bits. When this bit is reset to 0 and if the HUC/HMC bit is set, the frame is passed only if it matches the Hash filter.

[bit9] SAF (Source Address Filter)

The GMAC compares the SA field of the received frames with the values programmed in the enabled SA registers. When this bit is set to 1 and the SA filter fails, the GMAC drops the frame.
When this bit is reset to 0, then the GMAC forwards the received frame to the application.

[bit8] SAIF (Source Address Inverse Filter)

When this bit is set, the Address Check block operates in inverse filtering mode for the SA address comparison. The frames whose SA matches the SA registers will be marked as failing the SA Address filter.
When this bit is reset, frames whose SA does not match the SA registers will be marked as failing the SA Address filter.

[bit7, 6] PCF (Pass Control Frames)

These bits control the forwarding of all control frames (including unicast and multicast PAUSE frames).

00: GMAC filters all control frames from reaching the application

01: GMAC forwards all control frames except PAUSE control frames to application even if they fail the Address filter

10: GMAC forwards all control frames to application even if they fail the Address Filter

11: GMAC forwards control frames that pass the Address Filter.

The following conditions should be true for the PAUSE control frames processing:

Condition 1: The GMAC is in the full-duplex mode and flow control is enabled by setting the bit 2 (RFE) of the GMAC register 6 (Flow Control register) to 1.

Condition 2: The destination address (DA) of the received frame matches the special multicast address or the MAC Address 0 when bit 3 (UP) of the GMAC register 6 (Flow Control register) is set.

Condition 3: The Type field of the received frame is 0x8808 and the OPCODE field is 0x0001.

Note:

- *This field should be set to 01 only when the Condition 1 is true, that is, the GMAC is programmed to operate in the full-duplex mode and the RFE bit is enabled.*
  Otherwise, the PAUSE frame filtering may be inconsistent. When Condition 1 is false, the PAUSE frames are considered as generic control frames. Therefore, to pass all control frames (including PAUSE control frames) when full-duplex mode and flow control are not enabled, you should set the PCF field to 10 or 11 (as required by the application).

[bit5] DB (Disable Broadcast Frames)

When this bit is set to 1,the AFM module filters all incoming broadcast frames.
When this bit is reset to 0, the AFM module passes all received broadcast frames.

[bit4] PM (Pass All Multicast)

When this bit is set to "1", this bit indicates that all incoming frames with a multicast destination address (first bit in the destination address field is "1") are passed.
When this bit is reset to 0, filtering of multicast frame depends on HMC bit.

[bit3] DAIF (DA Inverse Filtering)

When this bit is set to 1, the Address Check block operates in inverse filtering mode for the DA address comparison for both unicast and multicast frames.
When this bit is reset to 0, normal filtering of frames is performed.
*This function is not valid when the frame is control frame.

[bit2] HMC (Hash Multicast)

When this bit is set to 1, GMAC performs destination address filtering of received multicast frames according to the hash table. When this bit is reset to 0, the GMAC performs a perfect destination address filtering for multicast frames, that is, it compares the DA field with the values programmed in DA registers.

[bit1] HUC (Hash Unicast)

When this bit is set to 1, GMAC performs destination address filtering of unicast frames according to the hash table.
When this bit is reset to 0, the GMAC performs a perfect destination address filtering for unicast frames, that is, it compares the DA field with the values programmed in DA registers.

[bit0] PR (Promiscuous Mode)

When this bit is set, the Address Filter module passes all incoming frames regardless of its destination or source address. The SA/DA Filter Fails status bits of the Receive Status Word will always be cleared when PR is set.

Table 4-2 and

Table 4-3 summarize the Destination and Source Address filtering based on the type of frames received.

Table 4-2 Destination Address Filtering Table

Frame Type	PR	HPF	HCU	DAIF	HMC	PM	DB	DA Filter Operation
Broadcast	1	X	X	X	X	X	X	Pass
	0	X	X	X	X	X	0	Pass
	0	X	X	X	X	X	1	Fail
Unicast	1	X	X	X	X	X	X	Pass all frames.
	0	X	0	0	X	X	X	Pass on Perfect/Group filter match.
	0	X	0	1	X	X	X	Fail on Perfect/Group filter match.
	0	0	1	0	X	X	X	Pass on Hash filter match.
	0	0	1	1	X	X	X	Fail on Hash filter match.
	0	1	1	0	X	X	X	Pass on Hash or Perfect/Group filter match.
	0	1	1	1	X	X	X	Fail on Hash or Perfect/Group filter match.
Multicast	1	X	X	X	X	X	X	Pass all frames.
	X	X	X	X	X	1	X	Pass all frames.
	0	X	X	0	0	0	X	Pass on Perfect/Group filter match and drop PAUSE control frames if PCF = 0x.
	0	0	X	0	1	0	X	Pass on Hash filter match and drop PAUSE control frames if PCF = 0x.
	0	1	X	0	1	0	X	Pass on Hash or Perfect/Group filter match and drop PAUSE control frames if PCF = 0x.
	0	X	X	1	0	0	X	Fail on Perfect/Group filter match and drop PAUSE control frames if PCF = 0x.
	0	0	X	1	1	0	X	Fail on Hash filter match and drop PAUSE control frames if PCF = 0x.
	0	1	X	1	1	0	X	Fail on Hash or Perfect/Group filter match and drop PAUSE control frames if PCF = 0x.

Table 4-3 Source Address Filtering Table

Frame Type	PR	SAIF	SAF	SA Filter Operation
Unicast	1	X	X	Pass all frames.
	0	0	0	Return pass status on Perfect/Group filter match but do not drop frames.
	0	1	0	Return fail status on Perfect/Group filter match but do not drop frame.
	0	0	1	Pass on Perfect/Group filter match and drop frames that fail.
	0	1	1	Fail on Perfect/Group filter match and drop frames that fail.

4.3 GMAC Register 2, 3 (MHTRH, MHTRL)

MHTRH and MHTRL registers are used to set the group address filtering.

Sets the upper 32 bits of the hash table in the HTH field.
Sets the lower 32 bits of the hash table in the HTL field.

MHTRH (MAC Hash Table register (High))	Address 0008h
bit	31 to 0
Field	HTH[31:0]
Attribute	R/W
Initial value	0

MHTRL (MAC Hash Table register (Low))	Address 000Ch
bit	31 to 0
Field	HTL[31:0]
Attribute	R/W
Initial value	0

[bit31:0] HTH, HTL

The 64-bit Hash table is used for group address filtering. For hash filtering, the contents of the destination address in the incoming frame is passed through the CRC logic, and the bit inversion value of the upper 6 bits of the CRC operation result are used to index the contents of the Hash table. The most significant bit of the index determines the register to be used (HTH/HTL), and the other 5 bits determine which bit within the register. A hash value of 00000 selects bit0 of the selected register, and a value of 11111 selects bit31 of the selected register.
If the corresponding bit value of the received register is 1, the received frame is accepted. Otherwise, it is rejected. If the PM (Pass All Multicast) bit is set in GMAC register 1, then all multicast frames are accepted regardless of the multicast hash values.
For example, if the DA of the incoming frame is received as 0x1F52419CB6AF (0x1F is the first byte received), then the internally calculated 6-bit Hash value is 0x2C and the HTH register bit12 is checked for filtering. If the DA of the incoming frame is received as 0xA00A98000045, then the calculated 6- bit Hash value is 0x07 and the HTL register bit7 is checked for filtering

The CRC32 is used

CRC_32:	$x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^{8}+x^{7}+x^{5}+x^{4}+x^{2}+x+1$
POLYNOMIAL:	0x04C11DB7
INITIAL_REMAINDER:	0xFFFFFFFF
FINAL_XOR_VALUE:	1’s complement of the CRC

Note:

- *Perform the light access to this register with the width of 32 bit.*
  To rewrite the different value to the register after a value is written, wait 4 cycle clocks of the PHY interface clock or more, and then rewrite the register.

4.4 GMAC Register 4 (GAR)

The GMII Address register controls the management cycles to the external PHY through the management interface.

GAR (GMII/MII Address register) Address 0010h

bit	31	30	29	28	27	26	25	24
Field	Reserved							
Attribute	R	R	R	R	R	R	R	R
Initial value	-	-	-	-	-	-	-	-

bit	23	22	21	20	19	18	17	16
Field	Reserved							
Attribute	R	R	R	R	R	R	R	R
Initial value	-	-	-	-	-	-	-	-

bit	15	14	13	12	11	10	9	8
Field	PA[4:0]					GR[4:2]		
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	GR[1:0]		CR[3:0]				GW	GB
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/WS_SC
Initial value	0	0	0	0	0	0	0	0

[bit15:11] PA (Physical Layer Address)

This field specifies the PHY address of the connected 32 PHY devices.

[bit10:6] GR (GMII Register)

These bits specify the accessed register address in the selected PHY device.

[bit5:2] CR (Application Clock Range)

The CR determines the frequency of the MDC clock as per the SYS_CLK frequency provided for Ethernet-MAC. The suggested range of SYS_CLK frequency applicable for each value below (when bit5 = 0) ensures that the MDC clock is approximately between the frequency range 1.25 MHz to 2.5 MHz.

CR	SYS_CLK	MDC Clock
0000	60 MHz -100 MHz	SYS_CLK/42
0001	100 MHz -150 MHz	SYS_CLK/62
0010	20 MHz – 35 MHz	SYS_CLK/16
0011	35 MHz – 60 MHz	SYS_CLK/26
0100	150 MHz – 250 MHz	SYS_CLK/102
0101	250 MHz – 300 MHz	SYS_CLK/122

Other than the above Reserved

[bit1] GW (GMII/MII Write)

When this bit is set to 1, this bit tells the PHY that this will be a Write operation using the value of the GMAC register5 (GMII Data). If this bit is reset to 0, this will be a Read operation, placing the data in the GMAC register5 (GMII Data).

[bit0] GB (GMII/MII Busy)

During a PHY register access, this bit will be set to 1 by the Application to indicate that a Read or Write access is in progress. This bit will be automatically set to 0 after the PHY register access is completed.

This bit should read a logic 0 before writing to GMAC register 4 and GMAC register 5.

During the PHY access, GMAC register4 and GMAC register5 should not be written to until this bit is cleared. The GMAC register 5(GMII Data) is invalid during a PHY Read operation.

4.5 GMAC Register 5 (GDR)

The GMII Data register sets Write data to be written to the PHY register and also stores Read data from the PHY register.

GDR (GMII/MII Data register) Address 0014h4h

bit	31	30	29	28	27	26	25	24
Field	Reserved							
Attribute	R	R	R	R	R	R	R	R
Initial value	-	-	-	-	-	-	-	-

bit	23	22	21	20	19	18	17	16
Field	Reserved							
Attribute	R	R	R	R	R	R	R	R
Initial value	-	-	-	-	-	-	-	-

bit	15	14	13	12	11	10	9	8
Field	GD[15:8]							
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	GD[7:0]							
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

[bit15:0] GD (GMII/MII Data Register)

This contains the 16-bit data value read from the PHY after a Management Read operation or the 16-bit data value to be written to the PHY before a Management Write operation.

4.6 GMAC Register 6 (FCR)

The Flow Control register controls the generation and reception of the Control (Pause Command) frames by the GMAC's Flow control module.

FCR (Flow Control register) Address 0018h

bit	31	30	29	28	27	26	25	24
Field	PT[15:8]							
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

bit	23	22	21	20	19	18	17	16
Field	PT[7:0]							
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

bit	15	14	13	12	11	10	9	8
Field	Reserved							
Attribute	R	R	R	R	R	R	R	R
Initial value	-	-	-	-	-	-	-	-

bit	7	6	5	4	3	2	1	0
Field	DZPQ	Reserved	PLT[1:0]		UP	RFE	TFE	FCB/BPA
Attribute	R/W	R	R/W	R/W	R/W	R/W	R/W	R/WS_SC / R/W
Initial value	0	-	0	0	0	0	0	0

A Write to a register with the Busy bit set to "1" triggers the Flow Control block to generate a Pause Control frame. The fields of the control frame are selected as specified in the 802.3x specification, and the Pause Time value from this register is used in the Pause Time field of the control frame. The Busy bit remains set until the control frame is transferred onto the cable. The Host must make sure that the Busy bit is cleared before writing to the register.

[bit31:16] PT (Pause Time)

This field holds the value to be used in the Pause Time field in the transmit control frame. The consecutive writes to this register should be performed only after at least 4 clock cycles.

[bit7] DZPQ (Disable Zero-Quanta Pause)

This bit is a reserved bit.

[bit5, 4] PLT (Pause Low Threshold)

This bit is a reserved bit.

[bit3] UP (Unicast Pause Frame detect)

When this bit is set to "1", GMAC will detect the Pause frames in the station's unicast address specified in MAC Address0 High register and MAC Address0 Low register, in addition to the detecting Pause frames with the unique multicast address. When this bit is reset, the GMAC will detect only a Pause frame with the unique multicast address specified in the 802.3x standard.

[bit2] RFE (Receive Flow Control Enable)

When this bit is set, the GMAC will decode the received Pause frame and disable its transmitter for a specified (Pause Time) time.
When this bit is reset, the decode function of the Pause frame is disabled.

[bit1] TFE (Transmit Flow Control Enable)

In Full-Duplex mode, when this bit is set to "1", GMAC enables the flow control operation to transmit Pause frames. When this bit is reset, the flow control operation in GMAC is disabled, and the GMAC will not transmit any Pause frames.
In Half-Duplex mode, when this bit is set, the GMAC enables the back-pressure operation. When this bit is reset, the backpressure feature is disabled.

[bit0] FCB/BPA (Flow Control Busy/Backpressure Activate)

This bit initiates a Pause Control frame in Full-Duplex mode and activates the backpressure function in Half-Duplex mode if TFE bit is set. In Full-Duplex mode, this bit should be read as 0 before writing to the Flow Control register. To initiate a Pause control frame, the Application must set this bit to 1. During a transmission of the Control Frame, this bit will continue to be set to signify that a frame transmission is in progress. After the completion of Pause control frame transmission, the GMAC will reset this bit to 0. The Flow Control register should not be written to until this bit is cleared.
In Half-Duplex mode, when this bit is set (and TFE is set), then backpressure is asserted by the F_GMAC4. During backpressure, when the GMAC receives a new frame, the transmitter starts sending a JAM pattern resulting in a collision. When the GMAC is configured to Half -Duplex mode, the BPA is automatically disabled.

4.7 GMAC Register 7 (VTR)

The VLAN Tag register sets a value to identify the VLAN frames.

VTR (VLAN TAG register) — Address 001Ch

bit	31	30	29	28	27	26	25	24
Field	Reserved							
Attribute	R	R	R	R	R	R	R	R
Initial value	-	-	-	-	-	-	-	-

bit	23	22	21	20	19	18	17	16
Field	Reserved							ETV
Attribute	R	R	R	R	R	R	R	R/W
Initial value	-	-	-	-	-	-	-	0

bit	15	14	13	12	11	10	9	8
Field	VL[15:8]							
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	VL[7:0]							
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

The GMAC compares the 13th and 14th bytes of the receiving frame (Length/Type) with 0x8100, and the following 2 bytes are compared with the VLAN tag; if a match occurs, it sets the received VLAN bit in the receive frame status. The legal length of the frame is increased from 1518 bytes to 1522 bytes.

[bit16] ETV (Enable 12-Bit VLAN Tag Comparison)

When this bit is set, a 12-bit VLAN identifier, rather than the complete 16-bit VLAN tag, is used for comparison and filtering. bit[11:0] of the VLAN tag are compared with the corresponding field in the received VLAN-tagged frame. When this bit is reset, all 16 bits of the received VLAN frame's fifteenth and sixteenth bytes are used for comparison.

[bit15:0] VL (VLAN Tag Identifier)

This contains the 802.1Q VLAN tag to identify VLAN frames, and is compared to the fifteenth and sixteenth bytes of the frames being received for VLAN frames. bit[15:13] are the User Priority, Bit[12] is the Canonical Format Indicator (CFI) and bits[11:0] are the VLAN tag's VLAN Identifier (VID) field.
When the ETV bit is set, only the VID (bit[11:0]) is used for comparison. If VL (VL[11:0] if ETV is set) is all zeros, the GMAC does not check the fifteenth and sixteenth bytes for VLAN tag comparison, and declares all frames with a Type field value of 0x8100 to be VLAN frames.

Note:

- *A write access to this register should be performed 32 bits long.*
  The consecutive writes to this register should be performed only after at least 4 clock cycles in the PHY interface clock.

4.8 GMAC Register 10 (RWFFR)

This register sets the Filter pattern of the Wake-up frame.

RWFFR (Remote Wake-up Frame Filter register) Address 0028h

bit	31	30	29	28	27	26	25	24
Field	RWFFR[31:24]							
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	-	-	-	-	-	-	-	-

bit	23	22	21	20	19	18	17	16
Field	RWFFR[23:16]							
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	-	-	-	-	-	-	-	-

bit	15	14	13	12	11	10	9	8
Field	RWFFR[15:8]							
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	RWFFR[7:0]							
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

[bit31:0] RWFFR [31:0] (Remote Wake-up Frame Filter Register)

This register sets the Filter pattern for the wake-up frame. 4 kinds of filter patterns can be programmable. The RWFFR consists of 8 kinds of registers as shown in the Figure 4-1.

The register is written by sequentially writing the eight register values into address 0x0028.
The register is read by sequentially loading the eight register values from address 0x0028.

The pointer of this register access is initialized by writing 1 to the RWFFRPR (bit 31 of the PMTR resister).

Figure 4-1 Register Configuration of RWFFR

RWFFR_0	Filter 0 Byte Mask							
RWFFR_1	Filter 1 Byte Mask							
RWFFR_2	Filter 2 Byte Mask							
RWFFR_3	Filter 3 Byte Mask							
RWFFR_4	Reserved	Filter 3 Command	Reserved	Filter 2 Command	Reserved	Filter 1 Command	Reserved	Filter 0 Command
RWFFR_5	Filter3 Offset		Filter2 Offset		Filter1 Offset		Filter0 Offset	
RWFFR_6	Filter 1 CRC-16				Filter 0 CRC-16			
RWFFR_7	Filter 3 CRC-16				Filter 2 CRC-16			

■ Filter i Byte Mask

This register defines which bytes of the frame are examined by filter i (0, 1, 2, and 3) in order to determine whether or not the frame is a wake-up frame. The MSB (thirty-first bit) must be zero. Bit j [30:0] is the Byte Mask. If bit j (byte number) of the Byte Mask is set, then Filter i Offset + j of the incoming frame is processed by the CRC block; otherwise Filter i Offset + j is ignored.

■ Filter i Command

This 4-bit command controls the filter i operation. Bit 3 specifies the address type, defining the pattern's destination address type. When the bit is set, the pattern applies to only multicast frames; when the bit is reset, the pattern applies only to unicast frame. Bit 2 and Bit 1 are reserved. Bit 0 is the enable for filter i; if Bit 0 is not set, filter i is disabled.

■ Filter i Offset

This register defines the offset (within the frame) from which the frames are examined by filter i. This 8-bit pattern-offset is the offset for the filter i first byte to examined. The minimum allowed is 12, which refers to the 13th byte of the frame (offset value 0 refers to the first byte of the frame).

■ Filter i CRC-16

This register contains the CRC_16 value calculated from the pattern, as well as the byte mask programmed to the wake-up filter register block.

CRC_16: $x^{16}+x^{15}+x^2+1$

POLYNOMIAL: 0x8005

INITIAL_REMAINDER: 0xFFFF

FINAL_XOR_VALUE: 0x0000

4.9 GMAC Register 11 (PMTR)

This register programs the wake-up request events and monitors the wake-up events.

PMTR (PMT register) Address 002Ch

bit	31	30	29	28	27	26	25	24
Field	RWFFRPR	Reserved						
Attribute	R/WS_SC	R	R	R	R	R	R	R
Initial value	-	-	-	-	-	-	-	-

bit	23	22	21	20	19	18	17	16
Field	Reserved							
Attribute	R	R	R	R	R	R	R	R
Initial value	-	-	-	-	-	-	-	-

bit	15	14	13	12	11	10	9	8
Field	Reserved						GU	Reserved
Attribute	R	R	R	R	R	R	R/W	R
Initial value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	Reserved	WPR	MPR	Reserved		WFE	MPE	PD
Attribute	R	R_SS_RC	R_SS_RC	R	R	R/W	R/W	R/WS_SC
Initial value	0	0	0	0	0	0	0	0

[bit31] RWFFRPR (Remote Wake-up Frame Filter Register Pointer Reset)

When this bit is set, resets the Remote Wake-up Frame Filter register pointer to 000. It is automatically cleared after 1 clock cycle.

[bit9] GU (Global Unicast)

When this bit is set, identifies any unicast packet which passes the DA filter to be a wake-up frame.

[bit6] WPR (Wake Up Frame Receive)

When this bit is set, this bit indicates the power management event was generated due to reception of a wake-up frame. This bit is cleared by a Read into this register.

[bit5] MPR (Magic Packet Received)

When this bit is set, this bit indicates the power management event was generated by the reception of a Magic Packet. This bit is cleared by a Read into this register.

[bit2] WFE (Wake-Up Frame Enable)

When this bit is set, enables generation of a power management event due to wake-up frame reception.

[bit1] MPE (Magic Packet Enable)

When this bit is set, enables generation of a power management event due to Magic Packet reception.

[bit0] PD (Power Down)

When this bit is set, all received frames will be dropped. This bit is cleared automatically when a Magic packet or Wake-Up frame is received, and Power-Down mode is disabled. Frames received after this bit is cleared are forwarded to the application. This bit must be set only when the Magic Packet Enable, Global, Unicast or Wake-Up Frame Enable bit is set high.

*: SYS_CLK can be stopped when power management mode. However, when the SYS_CLK is stopped, you cannot perform any read or write operations on this register. Therefore, the Software cannot clear this bit.

4.10 GMAC Register 12 (LPICSR)

This register controls the LPI functions and provides the LPI interrupt status.

LPICSR (LPI Control and Status register) Address 0030h

bit	31	30	29	28	27	26	25	24
Field	Reserved							
Attribute	R	R	R	R	R	R	R	R
Initial value	-	-	-	-	-	-	-	-

bit	23	22	21	20	19	18	17	16
Field	Reserved				LPITXA	PLSEN	PLS	LPIEN
Attribute	R	R	R	R	R/W	R/W	R/W	R/W
Initial value	-	-	-	-	0	0	0	0

bit	15	14	13	12	11	10	9	8
Field	Reserved						RLPIST	TLPIST
Attribute	R	R	R	R	R	R	R	R
Initial value	-	-	-	-	-	-	0	0

bit	7	6	5	4	3	2	1	0
Field	Reserved				RLPIEX	RLPIEN	TLPIEX	TLPIEN
Attribute	R	R	R	R	R_SS_RC	R_SS_RC	R_SS_RC	R_SS_RC
Initial value	0	0	0	0	0	0	0	0

[bit19] LPITXA (LPI TX Automate)

This bit controls the behavior of the GMAC when it is entering or coming out of the LPI mode on the transmit side.
If LPITXA and LPIEN bits are set to 1, the GMAC enters the LPI mode only after all outstanding frames and pending frames have been transmitted. The GMAC comes out of the LPI mode when the application sends any frames for transmission or the application issues a TX FIFO Flush command.
In addition, the GMAC automatically clears the LPIEN bit when it exits the LPI state.
When this bit is 0, the LPIEN bit directly controls behavior of the GMAC when it is entering or coming out of the LPI mode.

[bit18] PLSEN (PHY Link Status Enable)

This bit is reserved if the MII/RMII is selected. During writing to this bit, always write 0.

[bit17] PLS (PHY Link Status)

This bit writes the link status of the PHY. The GMAC Transmitter asserts the LPI pattern only when the link status is up (okay) at least for the time indicated by the LPI LS TIMER.
When this bit is set, the link is considered to be okay (up) and when reset, the link is considered to be down.

[bit16] LPIEN (LPI Enable)

When this bit is set, this bit instructs the GMAC Transmitter to enter the LPI state. When this bit is reset, this bit instructs the GMAC to exit the LPI state and resume normal transmission.

[bit9] RLPIST (Receive LPI State)

When this bit is set, this bit indicates that the GMAC is receiving LPI pattern on the PHY interface.

[bit8] TLPIST (Transmit LPI State)

When this bit is set, this bit indicates that the GMAC is transmitting LPI pattern on the PHY interface.

[bit3] RLPIEX (Receive LPI Exit)

When this bit is set, this bit indicates that the GMAC Receiver has stopped receiving the LPI pattern on PHY, has exited the LPI state, and has resumed the normal reception. This bit is cleared by a read into this register.

[bit2] RLPIEN (Receive LPI Entry)

When this bit is set, this bit indicates that the GMAC Receiver has received an LPI pattern and entered the LPI state. This bit is cleared by a read into this register.

Note:

- *When both of RLPIEN and RLPIEX is 1, you should assume Receive LPI is exit.*

[bit1] TLPIEX (Transmit LPI Exit)

When this bit is set, this bit indicates that the GMAC transmitter has exited the LPI state after the user has cleared the LPIEN bit and the LPI TW Timer has expired. This bit is cleared by a read into this register.

[bit0] TLPIEN (Transmit LPI Entry)

When this bit is set, this bit indicates that the GMAC Transmitter has entered the LPI state because of the setting of the LPIEN bit. This bit is cleared by a read into this register.

4.11 GMAC Register 13 (LPITCR)

This register controls the timeout values in LPI states. It specifies the time for which the GMAC transmits the LPI pattern and also the time for which the GMAC waits before resuming the normal transmission.

LPITCR (LPI Timers Control register) Address 0034h

bit	31	30	29	28	27	26	25	24
Field	Reserved						LIT[9:8]	
Attribute	R	R	R	R	R	R	R/W	R/W
Initial value	-	-	-	-	-	-	1	1

bit	23	22	21	20	19	18	17	16
Field	LIT[7:0]							
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	1	1	1	0	1	0	0	0

bit	15	14	13	12	11	10	9	8
Field	TWT[15:8]							
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	TWT[7:0]							
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

[bit25:16] LIT (LPI LS TIMER)

These bits specify the minimum time (in milliseconds) for which the link-status from the PHY should be up (okay) before the LPI pattern can be transmitted to the PHY. The GMAC does not transmit the LPI pattern even when LPIEN bit is set unless the LPI LS Timer reaches the programmed terminal count. The default value of the LPI LS Timer is 1000 (1 sec) as defined in the IEEE standard.

[bit15:0] TWT (LPI TW TIMER)

These bits specify the minimum time (in microseconds) for which the GMAC waits after it has stopped transmitting the LPI pattern to the PHY and before it resumes the normal transmission. The TLPIEX status bit is set after the expiry of this timer.

4.12 GMAC Register 14 (ISR)

This register indicates the interrupt status.

ISR (Interrupt Status register) Address 0038h

bit	31	30	29	28	27	26	25	24
Field	Reserved							
Attribute	R	R	R	R	R	R	R	R
Initial value	-	-	-	-	-	-	-	-

bit	23	22	21	20	19	18	17	16
Field	Reserved							
Attribute	R	R	R	R	R	R	R	R
Initial value	-	-	-	-	-	-	-	-

bit	15	14	13	12	11	10	9	8
Field	Reserved					LPIIS	TSIS	Reserved
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	COIS	TIS	RIS	MIS	PIS	Reserved	Reserved	RGIS
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

[bit10] LPIIS (LPI Interrupt Status)

This bit is set for any LPI state entry or exit in GMAC Transmitter or Receiver. This bit is cleared on reading the byte 0 of GMAC register 12 (LPI Control and Status register).

[bit9] TSIS (Time Stamp Interrupt Status)

This bit is set in the case as follows.

- The system time value equals or exceeds the value specified in the Target Time High and Low registers
- There is an overflow in the seconds register

This bit is cleared on reading the byte 0 of the "Time Stamp Status register (see "4.29. GMAC Register 458 (TSR)").

[bit7] COIS (MMC Receive Checksum Offload Interrupt Status)

This bit is set to 1 whenever an interrupt is generated in the GMAC register 130 (MMC Receive Checksum Offload Interrupt register). This bit is cleared when all the bits in this interrupt register are cleared.

[bit6] TIS (MMC Transmit Interrupt Status)

This bit is set to 1 whenever an interrupt is generated in the GMAC register 66 (MMC Transmit Interrupt register). This bit is cleared when all the bits in this interrupt register are cleared.

[bit5] RIS (MMC Receive Interrupt Status)

This bit is set to 1 whenever an interrupt is generated in the GMAC register65 (MMC Receive Interrupt register). This bit is cleared when all the bits in this interrupt register are cleared.

[bit4] MIS (MMC Interrupt Status)
This bit is set to 1 whenever any of COIS/TIS/RIS bit is set to 1 and cleared only when all of these bits are 0.

[bit3] PIS (PMT Interrupt Status)
This bit is set whenever a Magic packet or Wake-on-LAN frame is received in Power- Down mode (See bits 5 and 6 in the GMAC register 11(PMTR). This bit is cleared when both bits[6:5] are cleared due to a read operation to the PMT Control and Status register.

[bit0] RGIS (RGMII Interrupt Status)
This bit is reserved if the MII/RMII is selected.

4.13 GMAC Register 15 (IMR)

The Interrupt Mask register bit enables the user to mask the interrupt signal (INT_SBD) because of the corresponding event in the Interrupt Status register.

IMR (Interrupt Mask register) Address 003Ch

bit	31	30	29	28	27	26	25	24
Field	Reserved							
Attribute	R	R	R	R	R	R	R	R
Initial value	-	-	-	-	-	-	-	-

bit	23	22	21	20	19	18	17	16
Field	Reserved							
Attribute	R	R	R	R	R	R	R	R
Initial value	-	-	-	-	-	-	-	-

bit	15	14	13	12	11	10	9	8
Field	Reserved					LPIIM	TSIM	Reserved
Attribute	R	R	R	R	R	R/W	R/W	R
Initial value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	Reserved				PIM	Reserved	Reserved	RGIM
Attribute	R	R	R	R	R/W	R	R	R/W
Initial value	0	0	0	0	0	0	0	0

[bit10] LPIIM (LPI Interrupt Mask)

When this bit is set, this bit disables the assertion of the interrupt signal because of the setting of LPIIS (LPI Interrupt Status) bit in the GMAC register 14.

[bit9] TSIM (Time Stamp Interrupt Mask)

When this bit is set, this bit disables the assertion of the interrupt signal due to the setting of TSIS (Time Stamp Interrupt Status) bit in the GMAC register 14.

[bit3] PIM (PMT Interrupt Mask)

When this bit is set, this bit will disable the assertion of the interrupt signal due to the setting of PIS (PMT Interrupt Status) bit in the GMAC register 14.

[bit0] RGIM (RGMII Interrupt Mask)

This bit is reserved if the MII/RMII is selected.

4.14 GMAC Register 16 (MAR0H)

The MAC Address0 High register sets the upper 16 bits of the 6-byte first MAC address of the station.

MAR0H (MAC Address0 register (High)) Address 0040h

bit	31	30	29	28	27	26	25	24
Field	MO	Reserved						
Attribute	R	R	R	R	R	R	R	R
Initial value	1	-	-	-	-	-	-	-

bit	23	22	21	20	19	18	17	16
Field	Reserved							
Attribute	R	R	R	R	R	R	R	R
Initial value	-	-	-	-	-	-	-	-

bit	15	14	13	12	11	10	9	8
Field	A0[47:40]							
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	1	1	1	1	1	1	1	1

bit	7	6	5	4	3	2	1	0
Field	A0[39:32]							
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	1	1	1	1	1	1	1	1

Note that the first byte of transmission destination address that is received on the PHY interface corresponds to the LS Byte (bit[7:0]) of the MAC Address Low register. For example, if 0x11 :0x22:0x33:0x44:0x55:0x66 is received (0x11 is the first byte) on the PHY interface as the destination address, then the MAC Address0 register [47:0] is compared with 0x665544332211.

MAC Address[47:0] : UU:VV:WW:XX:YY:ZZ
[7:0] = UU, [15:8] = VV, [23:16] = WW, [31:24] = XX, [39:32] = YY, [47:40] = ZZ

MARL: 0xXXWWVVYYUU
MARH: 0x8000ZZYY

[bit31] MO (Must be one)

Always 1.

[bit15:0] A0[47:32] (MAC Address0[47:32])

This field sets the upper 16 bits [47:32] of the MAC address 0. This is used by the GMAC for filtering for received frames and for inserting the MAC address in the Transmit Flow Control (PAUSE) Frames.

Note:

- *A write access to this register should be performed by 32-bit long.*
  When the MAC address is set, the MAC address high register (MARxH) should be written. Then, the MAC address low register (MARxL) should be written.
  If the MAC address high register (MARxH) is only written, or the order of writing is reversed, the written value cannot be recognized correctly.

4.15 GMAC Register 17 (MAR0L)

The MAC Address0 Low register sets the lower 32 bits of the 6-byte first MAC address of the station.

MAR0L (MAC Address0 register (Low)) Address 0044h

bit	31	30	29	28	27	26	25	24
Field	A0[31:24]							
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	1	1	1	1	1	1	1	1

bit	23	22	21	20	19	18	17	16
Field	A0[23:16]							
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	1	1	1	1	1	1	1	1

bit	15	14	13	12	11	10	9	8
Field	A0[15:8]							
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	1	1	1	1	1	1	1	1

bit	7	6	5	4	3	2	1	0
Field	A0[7:0]							
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	1	1	1	1	1	1	1	1

[bit31:0] A0[31:0] (MAC Address0[31:0])

This field sets the lower 32 bits of the MAC address 0. This is used by the GMAC for filtering for received frames and for inserting the MAC address in the Transmit Flow Control (PAUSE) Frames.

Note:

- *A write access to this bit should be performed 32 bits long.*
  The consecutive writes to this register should be performed only after at least 4 clock cycles in the PHY interface clock.

4.16 GMAC Register 18, 20, 22, …, 542 (MAR1H, 2H, 3H, …,31H)

The MAC Address High register sets the upper 16 bits of the 6-byte second to thirty-second MAC address of the station.

MAR1H to MAR31H (MAC Address1 to 31 register -High)

Address 0048h,0050h,0058h,0060h,0068h,0070h,0078h,0080h,0088h,0090h,0098h,00A0h,00A8h, 00B0h,00B8h,0800h,0808h,0810h,0818h,0820h,0828h,0830h,0838h,0840h,0848h,0850h, 0858h,0860h,0868h,0870h,0878h

bit	31	30	29	28	27	26	25	24
Field	AE	SA	MBC					
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

bit	23	22	21	20	19	18	17	16
Field	Reserved							
Attribute	R	R	R	R	R	R	R	R
Initial value	-	-	-	-	-	-	-	-

bit	15	14	13	12	11	10	9	8
Field	A [47:40]							
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	1	1	1	1	1	1	1	1

bit	7	6	5	4	3	2	1	0
Field	A [39:32]							
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	1	1	1	1	1	1	1	1

[bit31] AE (Address Enable)

When this bit is set, the Address filter module uses the MAC addresses 1 to 31. When this bit is reset, the address filter module will ignore the MAC addresses 1 to 31.

[bit30] SA (Source Address)

When this bit is set, the MAC Address1[47:0] is used to compare with the SA fields of the received frame.
When this bit is reset, the MAC Address1[47:0] is used to compare with the DA fields of the received frame.

[bit29:24] MBC (Mask Byte Control)

These bits are mask control bits for comparison of each of the MAC Address bytes. When this bit is set to 1, the GMAC does not compare the corresponding byte of received DA/SA with the contents of MAC Address* registers. Each bit controls the masking of the bytes as follows:
bit29: MAC Address*[47:40]
bit28: MAC Address*[39:32]
…
bit24: MAC Address*[7:0]

[bit15:0] A

This field sets the upper 16 bits (47:32) of the 6-byte second to thirty-second MAC address.

Note:

- *A write access to this register should be performed by 32-bit long.*
  When the MAC address is set, the MAC address high register (MARxH) should be written. Then, the MAC address low register (MARxL) should be written.
  If the MAC address high register (MARxH) is only written, or the order of writing is reversed, the written value cannot be recognized correctly.

  Writing values to each field of AE, SA, and MBC cannot be valid by writing to the MAC address high register (MARxH) only.
  It will be valid after the MAC address low register (MARxL) is written.

4.17 GMAC Register 19, 21, 23, …, 543 (MAR1L, 2L, 3L, …,31L)

The MAC Address Low register (1 to 31) sets the lower 32 bits of the 6-byte second to thirty-second MAC address of the station.

MAR1L to MAR31L (n=1 to 31) (MAC Address1 to 31 register -Low)

Address 004Ch,0054h,005Ch,0064h,006Ch,0074h,007Ch,0084h,008Ch,0094h,009Ch,00A4h,00ACh, 00B4h,00BCh,0804h,080Ch,0814h,081Ch,0824h,082Ch,0834h,083Ch,0844h,084Ch,0854h, 085Ch,0864h,086Ch,0874h,087Ch

bit	31	30	29	28	27	26	25	24
Field	A[31:24]							
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	1	1	1	1	1	1	1	1

bit	23	22	21	20	19	18	17	16
Field	A[23:16]							
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	1	1	1	1	1	1	1	1

bit	15	14	13	12	11	10	9	8
Field	A[15:8]							
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	1	1	1	1	1	1	1	1

bit	7	6	5	4	3	2	1	0
Field	A[7:0]							
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	1	1	1	1	1	1	1	1

[bit31:0] A [31:0]

This field contains the lower 32 bits of the 6-byte second to thirty-second MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

Note:

- *A write access to this bit should be performed 32 bits long.*
  The consecutive writes to this register should be performed only after at least 4 clock cycles in the PHY interface clock.

4.18 GMAC Register 54 (RGSR)

The RGMII Status register indicates the status signals received by the RGMII from the PHY.

RGSR (RGMII Status register) Address 00D8h

bit	31	30	29	28	27	26	25	24
Field	Reserved							
Attribute	R	R	R	R	R	R	R	R
Initial value	-	-	-	-	-	-	-	-

bit	23	22	21	20	19	18	17	16
Field	Reserved							
Attribute	R	R	R	R	R	R	R	R
Initial value	-	-	-	-	-	-	-	-

bit	15	14	13	12	11	10	9	8
Field	Reserved							
Attribute	R	R	R	R	R	R	R	R
Initial value	-	-	-	-	-	-	-	-

bit	7	6	5	4	3	2	1	0
Field	Reserved				LS	LSP		LM
Attribute	R	R	R	R	R	R	R	R
Initial value	-	-	-	-	0	0	0	0

[bit3] LS (Link Status)

This bit is reserved if the MII/RMII is selected.

[bit2, 1] LSP (Link Speed)

These bits are reserved if the MII/RMII is selected.

[bit0] LM (Link Mode)

This bit is reserved if the MII/RMII is selected.

4.19 GMAC Register 448 (TSCR)

This register controls the operation of the System Time generator and the snooping of PTP packets for time-stamping in the Receiver.

TSCR (Time Stamp Control register) Address 0700h

bit	31	30	29	28	27	26	25	24
Field	Reserved							ATSFC
Attribute	R	R	R	R	R	R	R	R/WSC
Initial value	-	-	-	-	-	-	-	0

bit	23	22	21	20	19	18	17	16
Field	Reserved					TSENMF	TSPS	
Attribute	R	R	R	R	R	R/W	R/W	R/W
Initial value	-	-	-	-	-	0	0	0

bit	15	14	13	12	11	10	9	8
Field	TSMRM	TETSEM	TSIP4E	TSIP6E	TETSP	TSV2E	TSDB	TSEA
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	1	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	Reserved		TARU	TITE	TSU	TSI	TFCU	TSE
Attribute	R	R	R/WSC	R/WSC	R/WSC	R/WSC	R/W	R/W
Initial value	-	-	0	0	0	0	0	0

[bit24] ATSFC (Auxiliary Snapshot FIFO Clear)

This register is reserved. Write 0 to this bit in writing.

[bit18] TSENMF (Enable MAC address for PTP frame filtering)

When this bit is set, uses the DA MAC address (that matches any MAC Address registers 0 to 31) to filter the PTP frames when PTP is sent directly over Ethernet.

[bit17, 16] TSPS (SelectPTP packets for taking snapshots)

These bits along with TSMRM bit and TETSEM bit decide the set of PTP packet types for which snapshot needs to be taken. The encoding is given in Table 4-4.

[bit15] TSMRM (Enable Snapshot for Messages Relevant to Master)

When this bit is set, the snapshot is taken for messages relevant to master node only else snapshot is taken for messages relevant to slave node. This is valid only for ordinary clock and boundary clock node.

[bit14] TETSEM (Enable Time Stamp Snapshot for Event Messages)

When this bit is set, the time stamp snapshot is taken for event messages only (SYNC, Delay_Req, Pdelay_Req or Pdelay_Resp). When this bit is reset snapshot is taken for all other messages except Announce, Management and Signaling.

[bit13] TSIP4E (Enable Time Stamp Snapshot for IPv4 frames)

When this bit is set, the time stamp snapshot is taken for IPv4 frames.

[bit12] TSIP6E (Enable Time Stamp Snapshot for IPv6 frames)
When this bit is set, the time stamp snapshot is taken for IPv6 frames.

[bit11] TETSP (Enable Time Stamp Snapshot for PTP over Ethernet frames)
When this bit is set, the time stamp snapshot is taken for frames which have PTP messages in Ethernet frames (PTP over Ethernet) also. By default snapshots are taken for UDP-IP-Ethernet PTP packets.

[bit10] TSV2E (Enable PTP packet snooping for version 2 format)
When this bit is set, the PTP packets are snooped using the 1588 version 2 format else snooped using the version 1 format.

[bit9] TSDB (Time Stamp Digital or Binary rollover control)
When this bit is set, the Time Stamp Low register rolls over after 0x3B9A_C9FF value (i.e., 1 nanosecond accuracy) and increments the Time Stamp (High) seconds. When this bit is reset, the rollover value of sub-second register is 0x7FFF_FFFF. The sub-second increment has to be programmed correctly depending on the PTP reference clock frequency and this bit value.

[bit8] TSEA (Enable Time Stamp for All Frames)
When this bit is set, the time stamp snapshot is enabled for all frames received by the GMAC.

[bit5] TARU (Addend Register Update)
When this bit is set, the contents of the Time Stamp Addend register is updated in the PTP block for fine correction. This is cleared when the update is completed. This register bit should be zero before setting it. This is a reserved bit when only course correction option is selected.

[bit4] TITE (Time Stamp Interrupt Trigger Enable)
When this bit is set, the Time Stamp interrupt is generated when the System Time becomes greater than the value written in Target Time register. This bit is reset after the generation of Time Stamp Trigger Interrupt.

[bit3] TSU (Time Stamp Update)
When this bit is set, the system time is updated (added/subtracted) with the value specified in the System Time - Time Stamp High Update register and System Time -Time Stamp Low Update register. This register bit should be read zero before updating it. This bit is reset once the update is completed in hardware. The "Time Stamp Higher Word" register is not updated.

[bit2] TSI (Time Stamp Initialize)
When this bit is set, the system time is updated (over-written) with the value specified in the System Time - Time Stamp High Update register and System Time -Time Stamp Low Update register. This register bit should be read zero before updating it. This bit is reset once the initialize is complete. The "Time Stamp Higher Word" register can only be initialized.

[bit1] TFCU (Time Stamp Fine or Coarse Update)
When this bit is set, indicates that the system times update to be done using fine update method. When this bit is reset it indicates the system time stamp update to be done using Coarse method.

[bit0] TSE (Time Stamp Enable)
When this bit is set, the system time stamping is enabled for transmit and receive frames. When disabled timestamp is not generated for transmit and receive frames and the TimeStamp Generator is also suspended. User has to always initialize the TimeStamp (system time) after enabling this mode.

Table 4-4 Time-Stamp Snapshot Dependency on Register Bits

TSPS	TSMRM	TETSEM	Message for which snapshot is taken
00	X	0	SYNC, Follow_Up, Delay_Req, Delay_Resp
00	0	1	SYNC
00	1	1	Delay_Req
01	X	0	SYNC, Follow_Up, Delay_Req, Delay_Resp , PDelay_Req, PDelay_Resp, PDelay_Resp_Follow_Up
01	0	1	SYNC, PDelay_Req, PDelay_Resp
01	1	1	Delay_Req, Pdelay_Req, Pdelay_Resp
10	X	X	SYNC, Delay_Req
11	X	X	PDelay_Req, PDelay_Resp

4.20 GMAC Register 449 (SSIR)

In Coarse Update mode (TFCU bit in Time Stamp Control register), the value in this register is added to the system time every clock cycle of PTP_CLK. In Fine Update mode, the value in this register is added to the system time whenever the Accumulator gets an overflow.

SSIR (Sub-Second Increment register) Address 0704h

bit	31	30	29	28	27	26	25	24
Field	Reserved							
Attribute	R/WSC	R	R	R	R	R	R	R
Initial value	-	-	-	-	-	-	-	-

bit	23	22	21	20	19	18	17	16
Field	Reserved							
Attribute	R	R	R	R	R	R	R	R
Initial value	-	-	-	-	-	-	-	-

bit	15	14	13	12	11	10	9	8
Field	Reserved							
Attribute	R	R	R	R	R	R	R	R
Initial value	-	-	-	-	-	-	-	-

bit	7	6	5	4	3	2	1	0
Field	SSINC							
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

[bit7:0] SSINC (Sub-Second Increment Value)

The value programmed in this register is accumulated with the contents of the sub-second register. For example, to achieve an accuracy of 20 ns, the value to be programmed is 20. (0x14 with a 50 MHz reference clock if 1ns accuracy is selected.)

4.21 GMAC Register 450 (STSR)

The System Time - Seconds register, along with System Time - Nanoseconds register, indicates the current value of the system time maintained. Though it is updated on a continuous basis, there is some delay from the actual time due to clock domain transfer latencies (from PTP_CLK to SYS_CLK).

STSR (System Time - Seconds register) Address 0708h

bit	31	30	29	28	27	26	25	24
Field	TSS[31:24]							
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

bit	23	22	21	20	19	18	17	16
Field	TSS[23:16]							
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

bit	15	14	13	12	11	10	9	8
Field	TSS[15:8]							
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	TSS[7:0]							
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

[bit31:0] TSS (Time Stamp Second)

The value in this field indicates the current value in seconds of the System Time maintained.

4.22 GMAC Register 451 (STNR)

The System Time - Nanoseconds register uses along with System Time - Seconds register.

STNR (System Time - Nanoseconds register) Address 070Ch

bit	31	30	29	28	27	26	25	24
Field	Reserved	TSSS[30:24]						
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

bit	23	22	21	20	19	18	17	16
Field	TSSS[23:16]							
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

bit	15	14	13	12	11	10	9	8
Field	TSSS[15:8]							
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	TSSS[7:0]							
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

[bit30:0] TSSS (Time Stamp Sub-Seconds)

The value in this field has the sub second representation of time, with an accuracy of 0.46 ns. (When TSDB is set, each bit represents 1 ns and the maximum value will be 0x3B9A_C9FF, after which it rolls-over to zero).

4.23 GMAC Register 452 (STSUR)

The System Time - Seconds Update register, along with the System Time - Nanoseconds Update register, initialize or update the system time maintained. You must write both of these registers before setting the TSINIT or TSUPDT bits in the Time Stamp Control register.

STSUR (System Time - Seconds Update register) Address 0710h

bit	31	30	29	28	27	26	25	24
Field	TSS[31:24]							
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

bit	23	22	21	20	19	18	17	16
Field	TSS[23:16]							
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	

bit	15	14	13	12	11	10	9	8
Field	TSS[15:8]							
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	TSS[7:0]							
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

[bit31:0] TSS (Time Stamp Second)

The value in this field indicates the time, in seconds, to be initialized or added to the system time.

4.24 GMAC Register 453 (STNUR)

The System Time - Nanoseconds Update register uses along with System Time - Seconds Update register.

STSNUR (System Time - Nanoseconds Update register) Address 0714h

bit	31	30	29	28	27	26	25	24
Field	ADDSUB	TSSS[30:24]						
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

bit	23	22	21	20	19	18	17	16
Field	TSSS[23:16]							
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

bit	15	14	13	12	11	10	9	8
Field	TSSS[15:8]							
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	TSSS[7:0]							
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

[bit31] ADDSUB (Add or Subtract Time)

When this bit is set, the time value is subtracted with the contents of the update register. When this bit is reset, the time value is added with the contents of the update register.

[bit30:0] TSSS (Time Stamp Sub-Seconds)

The value in this field has the sub second representation of time, with an accuracy of 0.46 ns. (When TSDB is set in the time stamp control register, each bit represents 1 ns and the programmed maximum value should not exceed 0x3B9A_C9FF.)

4.25 GMAC Register 454 (TSAR)

This register value is used only when the system time is configured for Fine Update mode (TFCU bit in GMAC register 448). This register content is added to a 32-bit accumulator in every clock cycle (of PTP_CLK) and the system time is updated whenever the accumulator overflows.

TSAR (Time Stamp Addend register) Address 0718h

bit	31	30	29	28	27	26	25	24
Field	TSAR[31:24]							
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

bit	23	22	21	20	19	18	17	16
Field	TSAR[23:16]							
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

bit	15	14	13	12	11	10	9	8
Field	TSAR[15:8]							
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	TSAR[7:0]							
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

[bit31:0] TSAR (Time Stamp Addend Register)

This register indicates the 32-bit time value to be added to the Accumulator register to achieve time synchronization.

4.26 GMAC Register 455 (TTSR)

The Target Time Seconds register, along with Target Time Nanoseconds register, are used to schedule an interrupt event (TSTART bit in GMAC register 458, or otherwise, TSIS bit in GMAC register14[9]) when the system time exceeds the value programmed in these registers.

TTSR (Target Time Seconds register) Address 071Ch

bit	31	30	29	28	27	26	25	24
Field	TSTR[31:24]							
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

bit	23	22	21	20	19	18	17	16
Field	TSTR[23:16]							
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	

bit	15	14	13	12	11	10	9	8
Field	TSTR[15:8]							
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	TSTR[7:0]							
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

[bit31:0] TSTR (Target Time Stamp Seconds Register)

This register stores the time in seconds. When the time stamp value matches or exceeds both Target Time Stamp registers, the GMAC, if enabled, generates an interrupt.

4.27 GMAC Register 456 (TTNR)

The Target Time Nanoseconds register uses along with Target Time Seconds register.

TTNR (Target Time Nanoseconds register) Address 0720h

bit	31	30	29	28	27	26	25	24
Field	Reserved	TSTR[30:24]						
Attribute	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	-	0	0	0	0	0	0	0

bit	23	22	21	20	19	18	17	16
Field	TSTR[23:16]							
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

bit	15	14	13	12	11	10	9	8
Field	TSTR[15:8]							
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	TSTR[7:0]							
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

[bit30:0] TSTR (Target Time Stamp Nanoseconds Register)

This register stores the time in (signed) nanoseconds. When the value of the Time Stamp matches the Target time stamp registers (both), the GMAC will generate an interrupt if enabled. (This value should not exceed 0x3B9A_C9FF when TSDB is set in the time stamp control register.)

4.28 GMAC Register 457 (STHWSR)

This register reads the most significant 16-bit of the time stamp 48-bit seconds value.

STHWSR (System Time - Higher Word Seconds register) Address 0724h

bit	31	30	29	28	27	26	25	24
Field	Reserved							
Attribute	R	R	R	R	R	R	R	R
Initial value	-	-	-	-	-	-	-	-

bit	23	22	21	20	19	18	17	16
Field	Reserved							
Attribute	R	R	R	R	R	R	R	R
Initial value	-	-	-	-	-	-	-	-

bit	15	14	13	12	11	10	9	8
Field	TSHWR[15:8]							
Attribute	R/WSU	R/ WSU	R/ WSU	R/ WSU	R/ WSU	R/ WSU	R/ WSU	R/ WSU
Initial value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	TSHWR[7:0]							
Attribute	R/ WSU	R/ WSU	R/ WSU	R/ WSU	R/ WSU	R/ WSU	R/ WSU	R/ WSU
Initial value	0	0	0	0	0	0	0	0

[bit15:0] TSHWR (Time Stamp Higher Word Register)

Contains the most significant 16-bit of the time stamp seconds value. The register is directly written to initialize the value. This register is incremented when there is an overflow from the 32-bit of the System Time - Seconds register.

4.29 GMAC Register 458 (TSR)

This register indicates the operation status to the system time counter.

TSR (Time Stamp Status register) Address 0728h

bit	31	30	29	28	27	26	25	24
Field	Reserved				ATSNS			ATSSTM
Attribute	R	R	R	R	R	R	R	R_SS_RC
Initial value	-	-	-	-	0	0	0	0

bit	23	22	21	20	19	18	17	16
Field	Reserved							
Attribute	R	R	R	R	R	R	R	R
Initial value	-	-	-	-	-	-	-	-

bit	15	14	13	12	11	10	9	8
Field	Reserved							
Attribute	R	R	R	R	R	R	R	R
Initial value	-	-	-	-	-	-	-	-

bit	7	6	5	4	3	2	1	0
Field	Reserved				TRGTER	ATSTS	TSTART	TSSOVF
Attribute	R	R	R	R	R_SS_RC	R_SS_RC	R_SS_RC	R_SS_RC
Initial value	-	-	-	-	0	0	0	0

All bits except bit[27:25] gets cleared after this register is read by the host.

[bit27:25] ATSNS (Auxiliary Time Stamp Number of Snapshots)

These bits are reserved bits.

[bit24] ATSSTM (Auxiliary Time Stamp Snapshot Trigger Missed)

This bit is a reserved bit.

[bit3] TRGTER (Timestamp Target Time Error)

This bit is set when the target time, which is being programmed in Target Time registers (GMAC register 455(TTSR), GMAC register 456(TTNR)), has already elapsed. This bit is cleared when read by the application.

[bit2] ATSTS (Auxiliary Time Stamp Trigger Snapshot)

This bit is a reserved bit.

[bit1] TSTART (Time Stamp Target Time Reached)

When this bit is set, indicates the value of system time is greater or equal to the value specified in the Target Time Seconds and Nanoseconds registers

[bit0] TSSOVF (Time Stamp Seconds Overflow)

When this bit is set, indicates that the seconds value of the time stamp (when supporting version 2 format) has overflowed beyond 32'hFFFF_FFFF.

4.30 GMAC Register 459 (PPSCR)

This register controls the PTPPPS output pin.

PPSCR (PPS Control register) Address 072Ch

bit	31	30	29	28	27	26	25	24
Field	Reserved							
Attribute	R	R	R	R	R	R	R	R
Initial value	-	-	-	-	-	-	-	-

bit	23	22	21	20	19	18	17	16
Field	Reserved							
Attribute	R	R	R	R	R	R	R	R
Initial value	-	-	-	-	-	-	-	-

bit	15	14	13	12	11	10	9	8
Field	Reserved							
Attribute	R	R	R	R	R	R	R	R
Initial value	-	-	-	-	-	-	-	-

bit	7	6	5	4	3	2	1	0
Field	Reserved				PPSCTRL[3:0]			
Attribute	R	R	R	R	R/W	R/W	R/W	R/W
Initial value	-	-	-	-	0	0	0	0

[bit3:0] PPSCTRL (Controls the frequency of the PPS output)

These bits control the behavior of the PPS output (PTPPPS) signal. The default value of PPSCTRL is 0000 and the PPS output is 1 pulse (of width PTP_CLK) every second.

For other values of PPSCTRL, the PPS output becomes a generated clock of following frequencies:

PPSCTRL	Binary Rollover	Digital Rollover
0001	2 Hz	1 Hz
0010	4 Hz	2 Hz
0011	8 Hz	4 Hz
0100	16 Hz	8 Hz
…..		
1111	32.768 kHz	16.384 kHz

Notes:

In the binary rollover mode, the PPS output (PTPPPS) has a duty cycle of 50 percent with these frequencies.

In the digital rollover mode, the PPS output frequency is an average number. The actual clock is of different frequency that gets synchronized every second. For example:

- *When PPSCTRL = 0001, the PPS (1 Hz) has a low period of 537 ms and a high period of 463 ms*
- *When PPSCTRL = 0010, the PPS (2 Hz) is a sequence of:*
  - *One clock of 50 percent duty cycle and 537 ms period*
  - *Second clock of 463 ms period (268 ms low and 195 ms high)*

- *When PPSCTRL = 0011, the PPS (4 Hz) is a sequence of:*
  - *Three clocks of 50 percent duty cycle and 268 ms period*
  - *Fourth clock of 195 ms period (134 ms low and 61 ms high)*

This behavior is because of the non-linear toggling of the bits in the digital rollover mode in GMAC Register 451 (STNR).

4.31 GMAC Register 460(ATNR)

This register is a reserved register.

ATNR (Auxiliary Time Stamp - Nanoseconds register) Address 0730h

bit	31	30	29	28	27	26	25	24
Field	Reserved	ATN						
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

bit	23	22	21	20	19	18	17	16
Field	ATN							
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

bit	15	14	13	12	11	10	9	8
Field	ATN							
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	ATN							
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

This register is a reserved register.

4.32 GMAC Register 461(ATSR)

This register is a reserved register.

ATSR (Auxiliary Time Stamp - Seconds register) Address 0734h

bit	31	30	29	28	27	26	25	24
Field	ATS							
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

bit	23	22	21	20	19	18	17	16
Field	ATS							
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

bit	15	14	13	12	11	10	9	8
Field	ATS							
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	ATS							
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

This register is a reserved register.

4.33 DMA Register 0 (BMR)

This register sets the operation mode of the DMA.

BMR(Bus Mode register) Address 1000h

bit	31	30	29	28	27	26	25	24
Field	Reserved				TXPR	MB	AAL	8xPBL
Attribute	R	R	R	R	R/W	R/W	R/W	R/W
Initial value	-	-	-	-	0	0	0	0

bit	23	22	21	20	19	18	17	16
Field	USP	RPBL						FB
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	1	0

bit	15	14	13	12	11	10	9	8
Field	PR		PBL[5:0]					
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	1

bit	7	6	5	4	3	2	1	0
Field	ATDS	DSL[4:0]					DA	SWR
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/WS_SC
Initial value	0	0	0	0	0	0	0	1

[bit27] TXPR (Transmit Priority)

When set, this bit indicates that the transmit DMA has higher priority than the receive DMA during arbitration for the AHB master interface.

[bit26] MB (Mixed Burst)

When this bit is set to 1 and FB bit is 1, the AHB master interface will start all bursts of length more than 16 with INCR (undefined burst) whereas it will revert to fixed burst transfers (INCRx and SINGLE) for burst-length of 16 and below.

[bit25] AAL (Address-Aligned Beats)

This bit is reserved. Always write 0 to this bit.

[bit24] 8xPBL (8xPBL Mode)

When this bit is set to 1, this bit multiplies the PBL value programmed (bits [22:17] and bits [13:8]) eight times. Thus the DMA will transfer data in to a maximum of 8, 16, 32, 64 and 128 beats depending on the PBL value.

[bit23] USP (Use Separate PBL)

When this bit is set to 1, it configures the receive DMA to use the value configured in bits [22:17] as PBL while the PBL value in bits [13:8] is applicable to transmit DMA operations only. When this bit is reset to 0, the PBL value in bits [13:8] is applicable for both DMA engines.

[bit22:17] RPBL (RxDMA PBL)

This value specifies the maximum number of undefined burst (INCR) to be executed in one RxDMA transaction. The receive DMA will always attempt to burst as specified in RPBL each time it starts a Burst transfer on the host bus. RPBL can be programmed with permissible values of 1, 2, 4, 8, 16, and 32. Any other value will result in undefined behavior. These bits are valid and applicable only when USP is set to 1.

[bit16] FB (Fixed Burst)

This bit controls whether the AHB Master interface performs fixed burst transfers or not. When this bit is set, the AHB will use only SINGLE, INCR4, INCR8 or INCR16 during start of normal burst transfers. When this bit is reset, the AHB will use SINGLE and INCR burst transfer operations.

[bit15,14] PR (Rx:Tx priority ratio)

These bits controls the priority ratio in the weighted round-robin arbitration between the receive DMA and transmit DMA. These bits are valid only when bit 1 (DA: DMA Arbitration Scheme) is reset. The priority ratio is receive:transmit or transmit:receive depending on whether bit 27 (TXPR: Transmit Priority) is reset to 0 or set to 1.

00: 1:1
01: 2:1
10: 3:1
11: 4:1

[bit13:8] PBL (Programmable Burst Length)

This value specifies the maximum number of undefined burst (INCR) to be executed in one DMA transaction. The DMA will always attempt to burst as specified in PBL each time it starts a Burst transfer on the host bus. PBL can be programmed with permissible values of 1, 2, 4, 8, 16, and 32. Any other value will result in undefined behavior. When USP is set to 1, this PBL value is applicable for transmit DMA transactions only.
When 8xPBL is set, the maximum value of PBL is 16.
If the number of beats to be transferred is more than 33, then perform the following steps:

1. Set the 8xPBL mode.
2. Set the PBL.

For example, if the maximum number of beats to be transferred is 64, then first set 8xPBL to 1 and then set PBL to 8.

[bit7] ATDS(Alternate Descriptor Size)

When this bit is set to 1, the descriptor size is increased to 32 bytes(8 words). This is required when the Time-Stamp feature is enabled. When this bit is reset to 0, the descriptor size reverts back to 16 bytes (4 words). Then, a processing to TDES4-7 and RDES4-7 is not performed.

[bit6:2] DSL [4:0](Descriptor Skip Length)

This bit specifies the number of Word to skip between two unchained descriptors. The address skipping starts from the end of current descriptor to the start of next descriptor. When DSL value equals zero, then the descriptor table is taken as contiguous by the DMA, in Ring mode.

[bit1] DA (DMA Arbitration scheme)

- 0: Weighted Round-robin with Rx:Tx or Tx:Rx.
  The priority between the paths is according to the priority weights specified in PR and TXPR.
- 1: Fixed priority.
  Transmit DMA has priority over receive DMA when TXPR is set to 1. Otherwise, receive DMA has priority over transmit DMA.

[bit0] SWR (Software Reset)

When this bit is set to 1, the DMA Controller resets all internal registers and logic. It is cleared automatically after the reset operation has completed in all of the clock domains. Read a 0 value in this bit before re-programming any register.

Note:

- *The reset operation is completed only when all the resets in all the active clock domains are de-asserted. Hence it is essential that all the PHY inputs clocks (applicable for the selected PHY interface) are present for software reset completion.*

4.34 DMA Register 1 (TPDR)

The Transmit Poll Demand register enables the Transmit DMA to check whether or not the current descriptor is owned by DMA.

TPDR(Transmit Poll Demand register) Address 1004h

bit	31	30	29	28	27	26	25	24
Field	TPD[31:24]							
Attribute	R/WT	R/WT	R/WT	R/WT	R/WT	R/WT	R/WT	R/WT
Initial value	0	0	0	0	0	0	0	0

bit	23	22	21	20	19	18	17	16
Field	TPD[23:16]							
Attribute	R/WT	R/WT	R/WT	R/WT	R/WT	R/WT	R/WT	R/WT
Initial value	0	0	0	0	0	0	0	0

bit	15	14	13	12	11	10	9	8
Field	TPD[15:8]							
Attribute	R/WT	R/WT	R/WT	R/WT	R/WT	R/WT	R/WT	R/WT
Initial value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	TPD[7:0]							
Attribute	R/WT	R/WT	R/WT	R/WT	R/WT	R/WT	R/WT	R/WT
Initial value	0	0	0	0	0	0	0	0

The Transmit Poll Demand command is given to wake up the Transmit DMA if it is in Suspend mode. The Transmit DMA can go into Suspend mode due to an Underflow error in a transmitted frame or due to the unavailability of descriptors owned by Transmit DMA. You can give this command anytime and the Transmit DMA will reset this command once it starts re-fetching the current descriptor from host memory.

[bit31:0] TPD (Transmit Poll Demand)

When these bits are written with any value, the DMA reads the current descriptor pointed to by DMA register 18 (CHTDR). If that descriptor is not available (owned by Host), transmission returns to the Suspend state and DMA register 5[2] is asserted. If the descriptor is available, transmission resumes.

4.35 DMA Register 2 (RPDR)

The Receive Poll Demand register enables the receive DMA to check for new descriptors.

RPDR(Receive Poll Demand register) Address 1008h

bit	31	30	29	28	27	26	25	24
Field	RPD[31:24]							
Attribute	R/WT	R/WT	R/WT	R/WT	R/WT	R/WT	R/WT	R/WT
Initial value	0	0	0	0	0	0	0	0

bit	23	22	21	20	19	18	17	16
Field	RPD[23:16]							
Attribute	R/WT	R/WT	R/WT	R/WT	R/WT	R/WT	R/WT	R/WT
Initial value	0	0	0	0	0	0	0	0

bit	15	14	13	12	11	10	9	8
Field	RPD[15:8]							
Attribute	R/WT	R/WT	R/WT	R/WT	R/WT	R/WT	R/WT	R/WT
Initial value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	RPD[7:0]							
Attribute	R/WT	R/WT	R/WT	R/WT	R/WT	R/WT	R/WT	R/WT
Initial value	0	0	0	0	0	0	0	0

This command is given to wake up the receive DMA from Suspend state. The receive DMA can go into Suspend state only due to the unavailability of descriptors owned by it.

[bit31:0] RPD[31:0] (Receive Poll Demand)

When these bits are written with any value, the DMA reads the current descriptor pointed to by DMA register 19. If that descriptor is not available (owned by Host), reception returns to the Suspended state and DMA register 5[7] is asserted. If the descriptor is available, the Receive DMA returns to active state.

4.36 DMA Register 3 (RDLAR)

The Receive Descriptor List Address register points to the start of the Receive Descriptor List.

RDLAR(Receive Descriptor List Address register) Address 100Ch

bit	31	30	29	28	27	26	25	24
Field	SRL[31:24]							
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

bit	23	22	21	20	19	18	17	16
Field	SRL[23:16]							
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

bit	15	14	13	12	11	10	9	8
Field	SRL[15:8]							
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	SRL[7:2]						Must be 0	
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R	R
Initial value	0	0	0	0	0	0	0	0

The descriptor lists reside in the host's physical memory space and must be Word-aligned. The DMA internally converts it to bus width aligned address by making the corresponding LS bits to 0. Writing to DMA register 3 is permitted only when reception is stopped. When stopped, DMA register 3 must be written to before the receive Start command is given.

[bit31:2] SRL[31:2] (Start of Receive List)

This field contains the base address of the First Descriptor in the Receive Descriptor list.

[bit1:0] Must be 0

These bits, please be sure to write in "0."

4.37 DMA Register 4 (TDLAR)

The Transmit Descriptor List Address register points to the start of the Transmit Descriptor List.

TDLAR(Transmit Descriptor List Address register) Address 1010h

bit	31	30	29	28	27	26	25	24
Field	STL[31:24]							
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

bit	23	22	21	20	19	18	17	16
Field	STL[23:16]							
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

bit	15	14	13	12	11	10	9	8
Field	STL[15:8]							
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	STL[7:2]						Must be 0	
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R	R
Initial value	0	0	0	0	0	0	0	0

The descriptor lists reside in the host's physical memory space and must be Word -aligned. The DMA internally converts it to bus width aligned address by making the corresponding LS bits to 0. Writing to DMA register 4 is permitted only when transmission has stopped. When stopped, DMA register 4 can be written before the transmission Start command is given.

[bit31:2] STL (Start of Transmit List)

This field contains the base address of the First Descriptor in the Transmit Descriptor list.

[bit1:0] Must be 0

These bits, please be sure to write as "0."

4.38 DMA Register 5 (SR)

The Status register contains all the status bits that the DMA reports to the host.

SR(Status register)								Address 1014h
bit	31	30	29	28	27	26	25	24
Field	Reserved	GLPII	TTI	GPI	GMI	GLI	EB[2:1]	
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0
bit	23	22	21	20	19	18	17	16
Field	EB[0]	TS[2:0]			RS[2:0]			NIS
Attribute	R	R	R	R	R	R	R	R_SS_WC
Initial value	0	0	0	0	0	0	0	0
bit	15	14	13	12	11	10	9	8
Field	AIS	ERI	FBI	Reserved		ETI	RWT	RPS
Attribute	R_SS_WC	R_SS_WC	R_SS_WC	R	R	R_SS_WC	R_SS_WC	R_SS_WC
Initial value	0	0	0	0	0	0	0	0
bit	7	6	5	4	3	2	1	0
Field	RU	RI	UNF	OVF	TJT	TU	TPS	TI
Attribute	R_SS_WC	R_SS_WC	R_SS_WC	R_SS_WC	R_SS_WC	R_SS_WC	R_SS_WC	R_SS_WC
Initial value	0	0	0	0	0	0	0	0

DMA register 5 is usually read by the Software driver during an interrupt service routine or polling. Most of the fields in this register cause the host to be interrupted. DMA register 5 bits are not cleared when read. Writing 1 to (unreserved) bits in DMA register 5[16:0] clears them and writing 0 has no effect. Each field (bits[16:0]) can be masked by masking the appropriate bit in DMA register 7.

[bit30] GLPII (GMAC LPI Interrupt)

This bit indicates an interrupt event in the LPI logic of GMAC. The software must read the corresponding registers to get the exact cause of interrupt and clear its source to reset this bit to 0. The interrupt signal (INT_LPI) is asserted when this bit is 1.

[bit29] TTI (Time-Stamp Trigger Interrupt)

This bit indicates an interrupt event in the GMAC's Time Stamp Generator block. The software must read the corresponding registers to get the exact cause of interrupt and clear its source to reset this bit to 0. When this bit is 1, the interrupt signal (INT_SBD) is asserted.

[bit28] GPI (GMAC PMT Interrupt)

This bit indicates an interrupt event of Wake-up Frame or Magic Packet. The software must read the corresponding registers to get the exact cause of interrupt and clear its source to reset this bit to 0. The interrupt signal (INT_SBD and INT_PMT) is asserted when this bit is 1.

Note:

- *As the INT_PMT is generated in RX_CLK domain, its cleaning on a read to PMT Status register is not immediate. The resultant clear signal has to cross to the RX_CLK domain and then clear the interrupt source. This delay will be at least 4 clock cycles of RX_CLK and can be significant when the Ethernet-MAC is operating in 10Mbps mode.*

[bit27] GMI (GMAC MMC Interrupt)

This bit reflects an interrupt event in the MMC register. The software must read the corresponding registers to get the exact cause of interrupt and clear the source of interrupt to reset this bit to 0. The interrupt signal (INT_SBD) is asserted when this bit is 1.

[bit26] GLI (GMAC Line interface Interrupt)

This bit is a reserved bit.

[bit25:23] EB (Error Bits)

These bits indicate the type of error that caused a Bus Error at AHB bus access. Valid only with Fatal Bus Error bit (DMA register 5[13]) set. This field does not generate an interrupt.
bit23=1 : Error during data transfer by transmit DMA
bit23=0 : Error during data transfer by receive DMA

bit24=1 : Error during read transfer
bit24=0 : Error during write transfer

bit25=1 : Error during descriptor access
bit25=0 : Error during data buffer access

[bit22:20] TS (Transmit Process State)

These bits indicate the Transmit DMA state. This field does not generate an interrupt.
000: Stopped - Reset or Stop Transmit Command issued
001: Running - Fetching Transmit Transfer Descriptor
010: Running - Waiting for status
011: Running - Reading the data from memory buffer and queuing the data into the Transmit buffer
100: Time-Stamp write state
101: Reserved
110: Suspended – Transmit Buffer Underflow, or an Unavailable Transmit Descriptor
111: Running - Closing Transmit Descriptor

[bit19:17] RS (Receive Process State)

These bits indicate the Receive DMA state. This field does not generate an interrupt.
000: Stopped - Reset or Stop Receive Command issued
001: Running - Fetching Receive Transfer Descriptor
010: Reserved
011: Running - Waiting for receive packet
100: Suspended - Receive Descriptor Unavailable
101: Running - Closing Receive Descriptor
110: Time-Stamp write state
111: Running – Transferring the receive packet data from receive buffer to host memory.

[bit16] NIS (Normal Interrupt Summary)

Normal Interrupt Summary bit value is the logical OR of the following when the corresponding interrupt bits are enabled in DMA register 7:
Register 5[0]: Transmit Interrupt
Register 5[2]: Transmit Buffer Unavailable
Register 5[6]: Receive Interrupt
Register 5[14]: Early Receive Interrupt

Only unmasked bits affect the Normal Interrupt Summary bit.
This is a sticky bit and must be cleared (by writing 1 to this bit) each time a corresponding bit that causes NIS to be set is cleared.

[bit15] AIS (Abnormal Interrupt Summary)

Abnormal Interrupt Summary bit value is the logical OR of the following when the corresponding interrupt bits are enabled in DMA register 7:
Register 5[1]:Transmit Process Stopped
Register 5[3]:Transmit Jabber Timeout
Register 5[4]: Receive FIFO Overflow
Register 5[5]: Transmit Underflow
Register 5[7]: Receive Buffer Unavailable
Register 5[8]: Receive Process Stopped
Register 5[9]: Receive Watchdog Timeout
Register 5[10]: Early Transmit Interrupt
Register 5[13]: Fatal Bus Error

Only unmasked bits affect the Abnormal Interrupt Summary bit.
This is a sticky bit and must be cleared each time a corresponding bit that causes AIS to be set is cleared.

[bit14] ERI (Early Receive Interrupt)

This bit indicates that the DMA had filled the first data buffer of the packet. Receive Interrupt DMA register 5[6] automatically clears this bit.

[bit13] FBI (Fatal Bus Error Interrupt)

This bit indicates that a bus error occurred, as detailed in EB[2:0]. When this bit is set, the corresponding DMA engine disables all its bus accesses.

[bit10] ETI (Early Transmit Interrupt)

This bit indicates that the frame to be transmitted was fully transferred to the MTL Transmit FIFO.

[bit9] RWT (Receive Watchdog Timeout)

This bit is asserted when a frame with a length greater than 2,048 bytes is received (10,240 when Jumbo Frame mode is enabled).

[bit8] RPS (Receive process Stopped)

This bit is asserted when the Receive Process enters the Stopped state.

[bit7] RU (Receive Buffer Unavailable)
This bit indicates that the Next Descriptor in the Receive List is owned by the host and cannot be acquired by the DMA. Receive Process is suspended. To resume processing Receive descriptors, the host should change the ownership of the descriptor and issue a Receive Poll Demand command. If no Receive Poll Demand is issued, Receive frame transfer Process resumes when the next recognized incoming frame is received. DMA register 5[7] is set only when the previous Receive Descriptor was owned by the DMA.

[bit6] RI (Receive Interrupt)
This bit indicates the completion of frame reception. Specific frame status information has been posted in the descriptor. Reception remains in the Running state.

[bit5] UNF (Transmit underflow)
This bit indicates that the Transmit Buffer had an Underflow during frame transmission. Transmission is suspended and an Underflow Error TDES0[1] is set.

[bit4] OVF (Receive Overflow)
This bit indicates that the Receive Buffer had an Overflow during frame reception. If the partial frame is transferred to application, the overflow status is set in RDES0[11].

[bit3] TJT (Transmit Jabber Timeout)
This bit indicates that the Transmit Jabber Timer expired, meaning that the transmitter had been excessively active. The transmission process is aborted and placed in the Stopped state. This causes the Transmit Jabber Timeout TDES0[14] flag to assert.

[bit2] TU (Transmit Buffer Unavailable)
This bit indicates that the Next Descriptor in the Transmit List is owned by the host and cannot be acquired by the DMA. Transmission is suspended. bit[22:20] explain the Transmit Process state transitions. To resume processing transmit descriptors, the host should change the ownership of the descriptor and then issue a Transmit Poll Demand command.

[bit1] TPS (Transmit Process Stopped)
This bit is set when the transmission is stopped.

[bit0] TI (Transmit Interrupt)
This bit indicates that frame transmission is finished and TDES1[31] is set in the First Descriptor.

4.39 DMA Register 6 (OMR)

The Operation Mode register establishes the Transmit and Receive operating modes and commands. DMA register 6 should be written at the end of DMA initialization.

OMR(Operation Mode register) Address 1018h

bit	31	30	29	28	27	26	25	24
Field	Reserved					DT	RSF	DFF
Attribute	R	R	R	R	R	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

bit	23	22	21	20	19	18	17	16
Field	Reserved		TSF	FTF	Reserved			TTC[2]
Attribute	R	R	R/W	R/WS_SC	R	R	R	R/W
Initial value	-	-	0	0	0	0	0	0

bit	15	14	13	12	11	10	9	8
Field	TTC[1:0]		ST	Reserved				
Attribute	R/W	R/W	R/W	R	R	R	R	R
Initial value	0	0	0	-	-	-	-	-

bit	7	6	5	4	3	2	1	0
Field	FEF	FUF	Reserved	RTC[1:0]		OSF	SR	Reserved
Attribute	R/W	R/W	R	R/W	R/W	R/W	R/W	R
Initial value	0	0	0	0	0	0	0	0

[bit26] DT (Disable Dropping of TCP/IP Checksum Error Frames)

When this bit is set, the GMAC does not drop frames that have only errors detected by the Receive Checksum Offload engine. Such frames do not have any errors (including FCS error) in the Ethernet frame received by the GMAC but have errors in the encapsulated payload only. When this bit is reset, all error frames are dropped if the FEF bit is reset.

[bit25] RSF (Receive Store and Forward)

When this bit is set, the transfer to the host memory starts after all the receive frame data is written to MTL Receive FIFO, ignoring RTC bits. When this bit is reset, the receive FIFO operates in Cut-Through mode, the transfer to the host memory starts according to the threshold specified by the RTC bits.

[bit24] DFF (Disable Flushing of Received Frames)

When this bit is set, the receive DMA does not flush any frames due to the unavailability of receive descriptors/buffers as it does normally when this bit is reset.

[bit21] TSF (Transmit Store Forward)

When this bit is set, transmission to the PHY interface starts after a full frame is transferred to in the MTL Transmit FIFO. When this bit is set, the TTC values specified in DMA register 6[16:14] are ignored. This bit should be changed only when transmission is stopped.

[bit20] FTF (Flush Transmit FIFO)

When this bit is set, the transmit FIFO controller logic is reset to its default values and thus all data in the transmit FIFO is lost/flushed. This bit is cleared internally when the flushing operation is completed fully. The Operation Mode register should not be written to until this bit is cleared. The data that is already accepted by the GMAC transmitter will not be flushed. It will be scheduled for transmission and will result in underflow and runt frame transmission.

Note:

- *The flush operation completes only after emptying the transmit FIFO of its contents and all the pending Transmit Status of the transmitted frames are accepted by the host. In order to complete this flush operation, the PHY transmit clock (TX_CLK) is required to be active.*

[bit16:14] TTC[2:0] (Transmit Threshold Control)

These three bits control the threshold level of the MTL Transmit FIFO. Transmission to the PHY interface starts when the frame size within the MTL Transmit FIFO is larger than the threshold. In addition, full frames with a length less than the threshold are also transmitted. These bits are used only when the TSF bit (bit21) is reset.

000 : 64 Bytes
001 : 128 Bytes
010 : 192 Bytes
011 : 256 Bytes
100 : 40 Bytes
101 : 32 Bytes
110 : 24 Bytes
111 : 16 Bytes

[bit13] ST (Start/Stop Transmission Command)

When this bit is set, transmission DMA is placed in the Running state, and the DMA checks the Transmit List at the current position for a frame to be transmitted. Descriptor acquisition is attempted either from the current position in the list, which is the Transmit List Base Address set by DMA register 4, or from the position retained when transmission was stopped previously. If the current descriptor is not owned by the DMA, transmission enters the Suspended state and Transmit Buffer Unavailable (DMA register 5[2]) is set. The Start Transmission command is effective only when transmission is stopped. If the command is issued before setting DMA register 4, then the DMA behavior is unpredictable.
When this bit is reset, the transmission process is placed in the Stopped state after completing the transmission of the current frame. The Next Descriptor position in the Transmit List is saved, and becomes the current position when transmission is restarted. The stop transmission command is effective only when the transmission of the current frame is complete or when the transmission is in the Suspended state.

[bit7] FEF (Forward Error Frames)

When this bit is reset, the receive FIFO drops frames with error status (CRC error, collision error, GMII_ER, Giant frame, watchdog timeout, overflow). However, if the frame's start byte (write) pointer is already transferred to the read controller side (in Threshold mode), then the frames are not dropped.
When FEF is set, all frames except runt error frames are forwarded to the DMA. But when receive FIFO overflows when a partial frame is received, then such frames are dropped even when FEF is set.

[bit6] FUF (Forward Undersized Good Frames)

When this bit is set, the receive FIFO will forward Undersized frames (frames with no Error and length less than 64 bytes) including pad-bytes and CRC). When this bit is reset, the receive FIFO will drop all frames of less than 64 bytes, unless it is already transferred due to lower value of Receive Threshold.

[bit4,3] RTC (Receive Threshold Control)

These two bits control the threshold level of the MTL Receive FIFO. Transfer (request) to DMA starts when the frame size within the MTL Receive FIFO is larger than the threshold. In addition, full frames with a length less than the threshold are transferred automatically. These bits are valid only when the RSF bit is zero, and are ignored when the RSF bit is set to 1.

00: 64
01: 32
10: 96
11: 128

[bit2] OSF (Operate on Second Frame)

When this bit is set, this bit instructs the DMA to process a second frame of Transmit data even before Transmit status for first frame is obtained.

*To stop the transmit operation when this bit is set, run the following flow 1 or 2.

1. Disable the Transmit DMA and wait for any previous frame transmissions to complete (transmit FIFO empty). (This specification is applicable irrespective of OSF mode.)
2. Disable the GMAC transmitter by clearing the appropriate bits in the GMAC Configuration register. (There is no other way to know whether FIFO is empty or not. So you can wait till the STOP status and then you can disable the Transmitter.)

[bit1] SR (Start/Stop Receive)

When this bit is set, the DMA Receive process is placed in the Running state. The DMA attempts to acquire the descriptor from the Receive list and processes incoming frames. Descriptor acquisition is attempted from the current position in the list, which is the address set by DMA register 3 or the position retained when the Receive process was previously stopped. If no descriptor is owned by the DMA, reception is suspended and Receive Buffer Unavailable (DMA register 5[7]) is set. The Start Receive command is effective only when reception has stopped. If the command was issued before setting DMA register 3, DMA behavior is unpredictable.
When this bit is cleared, receive DMA operation is stopped after the transfer of the current frame. The next descriptor position in the transmit list is saved and becomes the current position after the Receive process is restarted. The Stop Receive command is effective only when the Receive process is in either the Running (waiting for receive packet) or in the Suspended state.

4.40 DMA Register 7 (IER)

This register enables an interrupt from the DMA.

IER(Interrupt Enable register) Address 101Ch

bit	31	30	29	28	27	26	25	24
Field	Reserved							
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

bit	23	22	21	20	19	18	17	16
Field	Reserved							NIE
Attribute	R	R	R	R	R	R	R	R/W
Initial value	0	0	0	0	0	0	0	0

bit	15	14	13	12	11	10	9	8
Field	AIE	ERE	FBE	Reserved		ETE	RWE	RSE
Attribute	R/W	R/W	R/W	R	R	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	RUE	RIE	UNE	OVE	TJE	TUE	TSE	TIE
Attribute	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

[bit16] NIE (Normal Interrupt Summary Enable)

When this bit is set, a normal interrupt is enabled. When this bit is reset, a normal interrupt is disabled. This bit enables the following bits:

DMA register 5[0]: Transmit Interrupt
DMA register 5[2]: Transmit Buffer Unavailable
DMA register 5[6]: Receive Interrupt
DMA register 5[14]: Early Receive Interrupt

[bit15] AIE (Abnormal Interrupt Summary Enable)

When this bit is set, an Abnormal Interrupt is enabled. When this bit is reset, an Abnormal Interrupt is disabled. This bit enables the following bits

DMA register 5[1]: Transmit Process Stopped
DMA register 5[3]: Transmit Jabber Timeout
DMA register 5[4]: Receive Overflow
DMA register 5[5]: Transmit Underflow
DMA register 5[7]: Receive Buffer Unavailable
DMA register 5[8]: Receive Process Stopped
DMA register 5[9]: Receive Watchdog Timeout
DMA register 5[10]: Early Transmit Interrupt
DMA register 5[13]: Fatal Bus Error

[bit14] ERE (Early Receive Interrupt Enable)
When this bit is set with Normal Interrupt Summary Enable (DMA register 7[16]), Early Receive Interrupt is enabled. When this bit is reset, Early Receive Interrupt is disabled.

[bit13] FBE (Fatal Bus Error Enable)
When this bit is set with Abnormal Interrupt Summary Enable (DMA register 7[15]), the Fatal Bus Error Interrupt is enabled. When this bit is reset, Fatal Bus Error Enable Interrupt is disabled.

[bit10] ETE (Early Transmit Interrupt Enable)
When this bit is set with an Abnormal Interrupt Summary Enable (DMA register 7[15]), Early Transmit Interrupt is enabled. When this bit is reset, Early Transmit Interrupt is disabled.

[bit9] RWE (Receive Watchdog Timeout Enable)
When this bit is set with Abnormal Interrupt Summary Enable (DMA register 7[15]), the Receive Watchdog Timeout Interrupt is enabled. When this bit is reset, Receive Watchdog Timeout Interrupt is disabled.

[bit8] RSE (Receive Process Stopped Enable)
When this bit is set with Abnormal Interrupt Summary Enable (DMA register 7[15]), Receive Stopped Interrupt is enabled. When this bit is reset, Receive Stopped Interrupt is disabled.

[bit7] RUE (Receive Buffer Unavailable Enable)
When this bit is set with Abnormal Interrupt Summary Enable (DMA register 7[15]), Receive Buffer Unavailable Interrupt is enabled. When this bit is reset, the Receive Buffer Unavailable Interrupt is disabled.

[bit6] RIE (Receive Interrupt Enable)
When this bit is set with Normal Interrupt Summary Enable (DMA register 7[16]), Receive Interrupt is enabled. When this bit is reset, Receive Interrupt is disabled.

[bit5] UNE (Transmit underflow Enable))
When this bit is set with Abnormal Interrupt Summary Enable (DMA register 7[15]), Transmit Underflow Interrupt is enabled. When this bit is reset, Underflow Interrupt is disabled.

[bit4] OVE(Receive Overflow Enable)
When this bit is set with Abnormal Interrupt Summary Enable (DMA register 7[15]), Receive Overflow Interrupt is enabled. When this bit is reset, Overflow Interrupt is disabled

[bit3] TJE (Transmit Jabber Timeout)
When this bit is set with Abnormal Interrupt Summary Enable (DMA register 7[15]), Transmit Jabber Timeout Interrupt is enabled. When this bit is reset, Transmit Jabber Timeout Interrupt is disabled.

[bit2] TUE (Transmit Buffer Unavailable)
When this bit is set with Normal Interrupt Summary Enable (DMA register 7[16]), Transmit Buffer Unavailable Interrupt is enabled. When this bit is reset, Transmit Buffer Unavailable Interrupt is disabled.

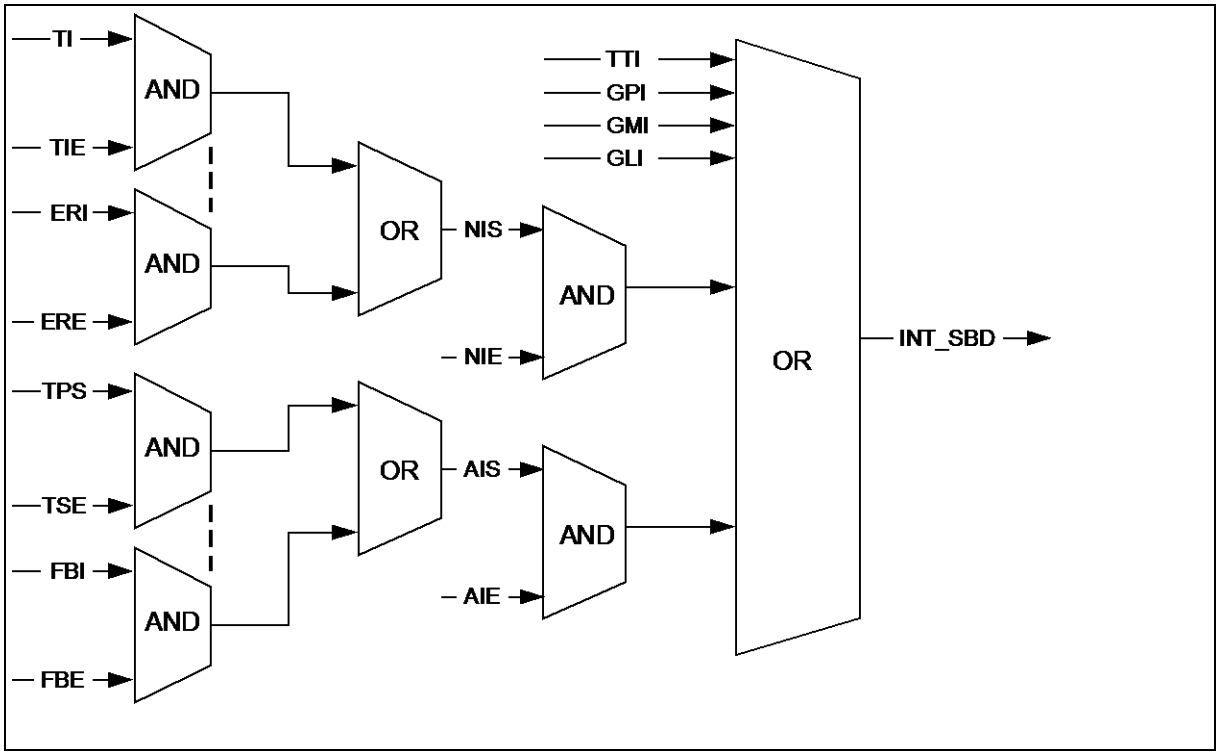
[bit1] TSE (Transmit Process Stopped)
When this bit is set with Abnormal Interrupt Summary Enable (DMA register 7[15]), Transmission Stopped Interrupt is enabled. When this bit is reset, Transmission Stopped Interrupt is disabled.

[bit0] TIE (Transmit Interrupt)

When this bit is set with Normal Interrupt Summary Enable (DMA register 7[16]), Transmit Interrupt is enabled. When this bit is reset, Transmit Interrupt is disabled.

The INT_SBD interrupt is generated as shown in Figure 4-2. It is asserted only when the TTI, GPI, GMI, or GLI bits of the DMA Status register is asserted, or when the NIS/AIS Status bit is asserted and the corresponding Interrupt Enable bits (NIE/AIE) are enabled.

Figure 4-2 INT_SBD Generation



4.41 DMA Register 8 (MFBOCR)

The DMA maintains two counters to track the number of missed frames during reception. This register reports the current value of the counter. The counter is used for diagnostic purposes.

MFBOCR (Missed Frame and Buffer Overflow Counter register) Address 1020h

bit	31	30	29	28	27	26	25	24
Field	Reserved			ONMFF	NMFF[10:7]			
Attribute	R	R	R	R_SS_RC	R_SS_RC	R_SS_RC	R_SS_RC	R_SS_RC
Initial value	0	0	0	0	0	0	0	0

bit	23	22	21	20	19	18	17	16
Field	NMFF[6:0]							ONMFH
Attribute	R_SS_RC	R_SS_RC	R_SS_RC	R_SS_RC	R_SS_RC	R_SS_RC	R_SS_RC	R_SS_RC
Initial value	0	0	0	0	0	0	0	0

bit	15	14	13	12	11	10	9	8
Field	NMFH[15:8]							
Attribute	R_SS_RC	R_SS_RC	R_SS_RC	R_SS_RC	R_SS_RC	R_SS_RC	R_SS_RC	R_SS_RC
Initial value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	NMFH[7:0]							
Attribute	R_SS_RC	R_SS_RC	R_SS_RC	R_SS_RC	R_SS_RC	R_SS_RC	R_SS_RC	R_SS_RC
Initial value	0	0	0	0	0	0	0	0

bit[15:0] indicate missed frames due to the host buffer being unavailable. bit[27:17] indicate missed frames due to buffer overflow conditions and runt frames (good frames of less than 64 bytes) dropped by the MTL.

[bit28] ONMFF (Overflow NMFF)

This bit indicates NMFF bit is overflow.

[bit27:17] NMFF (Number of Missed frame by Ethernet-MAC)

This bit indicates the number of frames missed by the application. The counter is cleared when this register is read.

[bit16] ONMFH (Overflow NMFH)

This bit indicates NMFH bit is overflow.

[bit15:0] NMFH[15:0] (Number of Missed frame by HOST)

This bit indicates the number of frames missed by the controller due to the Host Receive Buffer being unavailable. The counter is cleared when this register is read.

4.42 DMA Register 9 (RIWTR)

This register, when written with non-zero value, will enable the watchdog timer for RI (DMA register 5[6]).

RIWTR (Receive Interrupt Watchdog Timer register) Address 1024h

bit	31	30	29	28	27	26	25	24
Field	Reserved							
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

bit	23	22	21	20	19	18	17	16
Field	Reserved							
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

bit	15	14	13	12	11	10	9	8
Field	Reserved							
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	RIWT[7:0]							
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

[bit7:0] RIWT (RI Watchdog Timer count)

This bit sets the number of system cycles multiplied by 256 for which the watchdog timer is set. The watchdog timer gets triggered with the programmed time after the receive DMA completes the transfer of a frame for which the RI status bit is not set due to the setting in the corresponding receive descriptor RDES1[31]. When the watchdog timer runs out, the RI bit is set and the timer is stopped. The watchdog timer is reset when RI status bit is set to 1 due to setting of RDES1[31] of any received frame.

4.43 DMA Register 11 (AHBSR)

This register indicates the status of the AHB master interface.

AHBSR (AHB Status register) Address 102Ch

bit	31	30	29	28	27	26	25	24
Field	Reserved							
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

bit	23	22	21	20	19	18	17	16
Field	Reserved							
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

bit	15	14	13	12	11	10	9	8
Field	Reserved							
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	Reserved							AHBS
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

[bit0] AHBS (AHB Status)

When this bit is 1, it indicates that the AHB master interface is in the non-idle state.

4.44 DMA Register 18 (CHTDR)

The Current Host Transmit Descriptor register points to the start address of the current Transmit Descriptor read by the DMA.

CHTDR (Current Host Transmit Descriptor register) Address 1048h

bit	31	30	29	28	27	26	25	24
Field	HTDAP[31:24]							
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

bit	23	22	21	20	19	18	17	16
Field	HTDAP[23:16]							
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

bit	15	14	13	12	11	10	9	8
Field	HTDAP[15:8]							
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	HTDAP[7:0]							
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

[bit31:0] HTDAP[31:0] (Host Transmit Descriptor Address Pointer)

This bit is cleared on Reset. Pointer is updated by DMA during operation.

4.45 DMA Register 19 (CHRDR)

The Current Host Receive Descriptor register points to the start address of the current Receive Descriptor read by the DMA.

CHRDR (Current Host Receive Descriptor register) Address 104Ch

bit	31	30	29	28	27	26	25	24
Field	HRDAP[31:24]							
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

bit	23	22	21	20	19	18	17	16
Field	HRDAP[23:16]							
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

bit	15	14	13	12	11	10	9	8
Field	HRDAP[15:8]							
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	HRDAP[7:0]							
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

[bit31:0] HRDAP[31:0] (Host Receive Descriptor Address Pointer)

This bit is cleared on Reset. Pointer is updated by DMA during operation.

4.46 DMA Register 20 (CHTBAR)

The Current Host Transmit Buffer Address register points to the current Transmit Buffer Address being read by the DMA.

CHTBAR (Current Host Transmit Buffer Address register) Address 1050h

bit	31	30	29	28	27	26	25	24
Field	HTBAR[31:24]							
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

bit	23	22	21	20	19	18	17	16
Field	HTBAR[23:16]							
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

bit	15	14	13	12	11	10	9	8
Field	HTBAR[15:8]							
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	HTBAR[7:0]							
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

[bit31:0] HTBAR[31:0] (Host Transmit Buffer Address Register)

This bit is cleared on Reset. Pointer is updated by DMA during operation.

4.47 DMA Register 21 (CHRBAR)

The Current Host Receive Buffer Address register points to the current Receive Buffer address being read by the DMA.

CHRBAR (Current Host Receive Buffer Address register) Address 1054h

bit	31	30	29	28	27	26	25	24
Field	HRBAR[31:24]							
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

bit	23	22	21	20	19	18	17	16
Field	HRBAR[23:16]							
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

bit	15	14	13	12	11	10	9	8
Field	HRBAR[15:8]							
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	HRBAR[7:0]							
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

[bit31:0] HRBAR[31:0] (Host Receive Buffer Address Register)

This bit is cleared on Reset. Pointer is updated by DMA during operation.

4.48 MMC Register List

Table 4-5 shows the register map of the MMC module and the content of the statistics of each counter.

Table 4-5 MMC Register Map

Address	Register No.	Register	Description
0100	GMAC Reg. 64	mmc_cntl	MMC Control establishes the operating mode of MMC Counters. (see 4.49.)
0104	GMAC Reg. 65	mmc_intr_rx	MMC Receive Interrupt maintains the interrupt generated from all of the receive statistic counters. (see 4.50.)
0108	GMAC Reg. 66	mmc_intr_tx	MMC Transmit Interrupt maintains the interrupt generated from all of the transmit statistic counters. (see 4.51.)
010C	GMAC Reg. 67	mmc_intr_mask_rx	MMC Receive Interrupt mask maintains the mask for the interrupt generated from all of the receive statistic counters. (see 4.52.)
0110	GMAC Reg. 68	mmc_intr_mask_tx	MMC Transmit Interrupt mask maintains the mask for the interrupt generated from all of the transmit statistic counters. (see 4.53.)
0114	GMAC Reg. 69	txoctetcount_gb	Number of bytes transmitted, exclusive of preamble and retried bytes, in good and bad frames.
0118	GMAC Reg. 70	txframecount_gb	Number of good and bad frames transmitted, exclusive of retried frames.
011C	GMAC Reg. 71	txbroadcastframes_g	Number of good broadcast frames transmitted.
0120	GMAC Reg. 72	txmulticastframes_g	Number of good multicast frames transmitted.
0124	GMAC Reg. 73	tx64octets_gb	Number of good and bad frames transmitted with length of 64 bytes, exclusive of preamble and retried frames.
0128	GMAC Reg. 74	tx65to127octets_gb	Number of good and bad frames transmitted with length between 65 and 127 (inclusive) bytes, exclusive of preamble and retried frames.
012C	GMAC Reg. 75	tx128to255octets_gb	Number of good and bad frames transmitted with length between 128 and 255 (inclusive) bytes, exclusive of preamble and retried frames.
0130	GMAC Reg. 76	tx256to511octets_gb	Number of good and bad frames transmitted with length between 256 and 511 (inclusive) bytes, exclusive of preamble and retried frames.
0134	GMAC Reg. 77	tx512to1023octets_gb	Number of good and bad frames transmitted with length between 512 and 1023 (inclusive) bytes, exclusive of preamble and retried frames.
0138	GMAC Reg. 78	tx1024tomaxoctets_gb	Number of good and bad frames transmitted with length between 1024 and Maxsize (inclusive) bytes, exclusive of preamble and retried frames.
013C	GMAC Reg. 79	txunicastframes_gb	Number of good and bad unicast frames transmitted.
0140	GMAC Reg. 80	txmulticastframes_gb	Number of good and bad multicast frames transmitted.
0144	GMAC Reg. 81	txbroadcastframes_gb	Number of good and bad broadcast frames transmitted.
0148	GMAC Reg. 82	txunderflowerror	Number of frames aborted due to frame underflow error.
014C	GMAC Reg. 83	txsinglecol_g	Number of successfully transmitted frames after a single collision in Half-duplex mode.
0150	GMAC Reg. 84	txmulticol_g	Number of successfully transmitted frames after more than a single collision in Half-duplex mode.
0154	GMAC Reg. 85	txdeferred	Number of successfully transmitted frames after a deferral in Half-duplex mode.
0158	GMAC Reg. 86	txlatecol	Number of frames aborted due to late collision error.
015C	GMAC Reg. 87	txexesscol	Number of frames aborted due to excessive (16) collision errors.
0160	GMAC Reg. 88	txcarriererrror	Number of frames aborted due to carrier sense error (no carrier or loss of carrier).

Address	Register No.	Register	Description
0164	GMAC Reg. 89	txoctetcount_g	Number of bytes transmitted, exclusive of preamble, in good frames only.
0168	GMAC Reg. 90	txframecount_g	Number of good frames transmitted.
016C	GMAC Reg. 91	txexecessdef	Number of frames aborted due to excessive deferral error (deferred for more than two max-sized frame times).
0170	GMAC Reg. 92	txpauseframes	Number of good PAUSE frames transmitted.
0174	GMAC Reg. 93	txvlanframes_g	Number of good VLAN frames transmitted, exclusive of retried frames.
0178,017C	Reserved	Reserved	-
0180	GMAC Reg. 96	rxframecount_gb	Number of good and bad frames received.
0184	GMAC Reg. 97	rxoctetcount_gb	Number of bytes received, exclusive of preamble, in good and bad frames.
0188	GMAC Reg. 98	rxoctetcount_g	Number of bytes received, exclusive of preamble, only in good frames.
018C	GMAC Reg. 99	rxbroadcastframes_g	Number of good broadcast frames received.
0190	GMAC Reg. 100	rxmulticastframes_g	Number of good multicast frames received.
0194	GMAC Reg. 101	rxcrcerror	Number of frames received with CRC error.
0198	GMAC Reg. 102	rxallignmenterror	Number of frames received with alignment (dribble) error. Valid only in 10/100 mode.
019C	GMAC Reg. 103	rxrunterror	Number of frames received with runt (<64 bytes and CRC error) error.
01A0	GMAC Reg. 104	rxjabbererror	Number of frames received with length greater than 1518 bytes with CRC error.
01A4	GMAC Reg. 105	rxundersize_g	Number of frames received with length less than 64 bytes, without any errors.
01A8	GMAC Reg. 106	rxoversize_g	Number of frames received with length greater than the maxsize without error.
01AC	GMAC Reg. 107	rx64octets_gb	Number of good and bad frames received with length 64 bytes, exclusive of preamble.
01B0	GMAC Reg. 108	rx65to127octets_gb	Number of good and bad frames received with length between 65 and 127 (inclusive) bytes, exclusive of preamble.
01B4	GMAC Reg. 109	rx128to255octets_gb	Number of good and bad frames received with length between 128 and 255 (inclusive) bytes, exclusive of preamble.
01B8	GMAC Reg. 110	rx256to511octets_gb	Number of good and bad frames received with length between 256 and 511 (inclusive) bytes, exclusive of preamble.
01BC	GMAC Reg. 111	rx512to1023octets_gb	Number of good and bad frames received with length between 512 and 1023 (inclusive) bytes, exclusive of preamble.
01C0	GMAC Reg. 112	rx1024tomaxoctets_gb	Number of good and bad frames received with length between 1024 and maxsize (inclusive) bytes, exclusive of preamble.
01C4	GMAC Reg. 113	rxunicastframes_g	Number of good unicast frames received.
01C8	GMAC Reg. 114	rxlengtherror	Number of frames received with length error (Length type field is not the frame size), for all frames with valid length field.
01CC	GMAC Reg. 115	rxoutofrangetype	Number of frames received with length/type field not equal to the valid frame size (>1500)
01D0	GMAC Reg. 116	rxpauseframes	Number of good and valid PAUSE frames received.
01D4	GMAC Reg. 117	rxfifooverflow	Number of missed received frames due to FIFO overflow.
01D8	GMAC Reg. 118	rxvlanframes_gb	Number of good and bad VLAN frames received.
01DC	GMAC Reg. 119	rxwatchdogerror	Number of frames received with error due to watchdog timeout error (frames with a data load larger than 2048 bytes).

Address	Register No.	Register	Description
01E0-01FC	Reserved	Reserved	-
0200	GMAC Reg. 128	mmc_ipc_intr_mask_rx	MMC IPC Receive Checksum Offload Interrupt Mask maintains the mask for the interrupt generated from the receive IPC statistic counters. (see 4.54.)
0204	Reserved	Reserved	-
0208	GMAC Reg. 130	mmc_ipc_intr_rx	MMC Receive Checksum Offload Interrupt maintains the interrupt that the receive IPC statistic counters generate. (see 4.55.)
020C	Reserved	Reserved	-
0210	GMAC Reg. 132	rxipv4_gd_frms	Number of good IPv4 datagrams received with the TCP, UDP, or ICMP payload
0214	GMAC Reg. 133	rxipv4_hdrerr_frms	Number of IPv4 datagrams received with header errors (checksum, length, or version mismatch)
0218	GMAC Reg. 134	rxipv4_nopay_frms	Number of IPv4 datagram frames received that did not have a TCP, UDP, or ICMP payload processed by the Checksum engine
021C	GMAC Reg. 135	rxipv4_frag_frms	Number of good IPv4 datagrams with fragmentation
0220	GMAC Reg. 136	rxipv4_udsbl_frms	Number of good IPv4 datagrams received that had a UDP payload with checksum disabled
0224	GMAC Reg. 137	rxipv6_gd_frms	Number of good IPv6 datagrams received with TCP, UDP, or ICMP payloads
0228	GMAC Reg. 138	rxipv6_hdrerr_frms	Number of IPv6 datagrams received with header errors (length or version mismatch)
022C	GMAC Reg. 139	rxipv6_nopay_frms	Number of IPv6 datagram frames received that did not have a TCP, UDP, or ICMP payload. This includes all IPv6 datagrams with fragmentation or security extension headers
0230	GMAC Reg. 140	rxudp_gd_frms	Number of good IP datagrams with a good UDP payload. This counter is not updated when the rxipv4_udsbl_frms counter is incremented.
0234	GMAC Reg. 141	rxudp_err_frms	Number of good IP datagrams whose UDP payload has a checksum error
0238	GMAC Reg. 142	rxtcp_gd_frms	Number of good IP datagrams with a good TCP payload
023C	GMAC Reg. 143	rxtcp_err_frms	Number of good IP datagrams whose TCP payload has a checksum error
0240	GMAC Reg. 144	rxicmp_gd_frms	Number of good IP datagrams with a good ICMP payload
0244	GMAC Reg. 145	rxicmp_err_frms	Number of good IP datagrams whose ICMP payload has a checksum error
0248, 024C	Reserved	Reserved	-
0250	GMAC Reg. 148	rxipv4_gd_octets	Number of bytes received in good IPv4 datagrams encapsulating TCP, UDP, or ICMP data. (Ethernet header, FCS, pad, or IP pad bytes are not included in this counter or in the octet counters listed below).
0254	GMAC Reg. 149	rxipv4_hdrerr_octets	Number of bytes received in IPv4 datagrams with header errors (checksum, length, version mismatch). The value in the Length field of IPv4 header is used to update this counter.
0258	GMAC Reg. 150	rxipv4_nopay_octets	Number of bytes received in IPv4 datagrams that did not have a TCP, UDP, or ICMP payload. The value in the IPv4 header's Length field is used to update this counter.
025C	GMAC Reg. 151	rxipv4_frag_octets	Number of bytes received in fragmented IPv4 datagrams. The value in the IPv4 header's Length field is used to update this counter.

Address	Register No.	Register	Description
0260	GMAC Reg. 152	rxipv4_udsbl_octets	Number of bytes received in a UDP segment that had the UDP checksum disabled. This counter does not count IP Header bytes.
0264	GMAC Reg. 153	rxipv6_gd_octets	Number of bytes received in good IPv6 datagrams encapsulating TCP, UDP or ICMPv6 data
0268	GMAC Reg. 154	rxipv6_hdrerr_octets	Number of bytes received in IPv6 datagrams with header errors (length, version mismatch). The value in the IPv6 header's Length field is used to update this counter.
026C	GMAC Reg. 155	rxipv6_nopay_octets	Number of bytes received in IPv6 datagrams that did not have a TCP, UDP, or ICMP payload. The value in the IPv6 header's Length field is used to update this counter.
0270	GMAC Reg. 156	rxudp_gd_octets	Number of bytes received in a good UDP segment. This counter (and the counters below) does not count IP header bytes.
0274	GMAC Reg. 157	rxudp_err_octets	Number of bytes received in a UDP segment that had checksum errors
0278	GMAC Reg. 158	rxtcp_gd_octets	Number of bytes received in a good TCP segment
027C	GMAC Reg. 159	rxtcp_err_octets	Number of bytes received in a TCP segment with checksum errors
0280	GMAC Reg. 160	rxicmp_gd_octets	Number of bytes received in a good ICMP segment
0284	GMAC Reg. 161	rxicmp_err_octets	Number of bytes received in an ICMP segment with checksum errors
0288-02FC	Reserved	Reserved	-

Note:

- *The following address of MMC counter register except MMC Control register, MMC Receive Interrupt register, MMC receive Interrupt Register, MMC Receive Checksum Offload Interrupt Mask register, MMC Receive Interrupt Mask register, MMC Transmit Interrupt Mask register, and MMC Receive Checksum Offload Interrupt Mask register is read only and the initial value is 0.*
- *0x0114 to 0x0174*
- *0x0180 to 0x01DC*
- *0x0210 to 0x0244*
- *0x0250 to 0x0284*

4.49 GMAC Register.64 (mmc_cntl)

The MMC Control register establishes the operating mode of the management counters.

Table 4-6 MMC Control Register (mmc_cntl) Address:0100h

bit	R/W Permission	Initial value	Description
31:6	R	0	Reserved
5	R/W	0	Full-Half preset When this bit is 0 and bit4 is set, all MMC counters get preset to almost-half value. All octet counters get preset to 0x7FFF_F800 (half - 2KBytes) and all frame-counters gets preset to 0x7FFF_FFF0 (half - 16) When this bit is 1 and bit4 is set, all MMC counters get preset to almost-full value. All octet counters get preset to 0xFFFF_F800 (full - 2KBytes) and all frame-counters gets preset to 0xFFFF_FFF0 (full - 16)
4	R/W_SC	0	Counters Preset When this bit is set, all counters will be initialized or preset to almost full or almost half as per Bit5 above. This bit will be cleared automatically after 1 clock cycle. This bit along with bit5 is useful for debugging and testing the assertion of interrupts due to MMC counter becoming half-full or full.
3	R/W	0	MMC Counter Freeze When this bit is set, this bit freezes all the MMC counters to their current value. (None of the MMC counters are updated due to any transmitted or received frame until this bit is reset to 0. If any MMC counter is read with the Reset on Read of bit2 set, then that counter is also cleared in this mode.)
2	R/W	0	Reset on read When this bit is set, the MMC counters will be reset to zero after Read (self-clearing after reset). The counters are cleared when the least significant byte lane (bits[7:0]) is read.
1	R/W	0	Counter Stop Rollover When this bit is set, counter after reaching maximum value will not roll over to zero.
0	R/W_SC	0	Counter reset When this bit is set, all counters will be reset. This bit will be cleared automatically after 1 clock cycle.

4.50 GMAC Register.65 (mmc_intr_rx)

This register indicates an interrupt from each receive statistic counter.

Table 4-7 MMC Receive Interrupt Register (mmc_intr_rx) Address:0104h

bit	R/W Permission	Initial value	Description
31:24	R	0	Reserved
23	R_SS_RC	0	This bit is set to "1" when the rxwatchdogerror counter reaches half the maximum value and also when it reaches the maximum value.
22	R_SS_RC	0	This bit is set to "1" when the rxvlanframes_gb counter reaches half the maximum value and also when it reaches the maximum value.
21	R_SS_RC	0	This bit is set to "1" when the Rxfifooverflow_gb counter reaches half the maximum value and also when it reaches the maximum value.
20	R_SS_RC	0	This bit is set to "1" when the rxpauseframes counter reaches half the maximum value and also when it reaches the maximum value.
19	R_SS_RC	0	This bit is set to "1" when the rxoutofrangetype counter reaches half the maximum value and also when it reaches the maximum value.
18	R_SS_RC	0	This bit is set to "1" when the Rxlengtherror counter reaches half the maximum value and also when it reaches the maximum value.
17	R_SS_RC	0	This bit is set to "1" when the rxunicastframes_gb counter reaches half the maximum value and also when it reaches the maximum value.
16	R_SS_RC	0	This bit is set to "1" when the rx1024tomaxoctets_gb counter reaches half the maximum value and also when it reaches the maximum value.
15	R_SS_RC	0	This bit is set to "1" when the rx512to1023octets_gb counter reaches half the maximum value and also when it reaches the maximum value.
14	R_SS_RC	0	This bit is set to "1" when the rx256to511octets_gb counter reaches half the maximum value and also when it reaches the maximum value.
13	R_SS_RC	0	This bit is set to "1" when the rx128to255maxoctects_gb counter reaches half the maximum value and also when it reaches the maximum value.
12	R_SS_RC	0	This bit is set to "1" when the rx65to127maxoctets_gb counter reaches half the maximum value and also when it reaches the maximum value.
11	R_SS_RC	0	This bit is set to "1" when the rxto64octets_gb counter reaches half the maximum value and also when it reaches the maximum value.
10	R_SS_RC	0	This bit is set to "1" when the rxoversize_g counter reaches half the maximum value and also when it reaches the maximum value.
9	R_SS_RC	0	This bit is set to "1" when the rxundersize_g counter reaches half the maximum value and also when it reaches the maximum value.
8	R_SS_RC	0	This bit is set to "1" when the Rxjabbererror counter reaches half the maximum value and also when it reaches the maximum value.
7	R_SS_RC	0	This bit is set to "1" when the rxrunterror counter reaches half the maximum value and also when it reaches the maximum value.
6	R_SS_RC	0	This bit is set to "1" when the rxallignmenterror counter reaches half the maximum value and also when it reaches the maximum value.
5	R_SS_RC	0	This bit is set to "1" when the rxcrcerror counter reaches half the maximum value and also when it reaches the maximum value.

bit	R/W Permission	Initial value	Description
4	R_SS_RC	0	This bit is set to "1" when the rxmulticastframes_g counter reaches half the maximum value and also when it reaches the maximum value.
3	R_SS_RC	0	This bit is set to "1" when the rxbroadcastframes_g counter reaches half the maximum value and also when it reaches the maximum value.
2	R_SS_RC	0	This bit is set to "1" when the rxoctetcounter_g counter reaches half the maximum value and also when it reaches the maximum value.
1	R_SS_RC	0	This bit is set to "1" when the rxoctetcounter_gb counter reaches half the maximum value and also when it reaches the maximum value.
0	R_SS_RC	0	This bit is set to "1" when the Rxframecount_gb counter reaches half the maximum value and also when it reaches the maximum value.

Each bit of the MMC Receive Interrupt register maintains the interrupts generated when receive statistic counters reach half their maximum values (0x8000_0000 for 32-bit counter and 0x8000 for 16-bit counter), and when they reach their maximum values (0xFFFF_FFFF for 32-bit counter and 0xFFFF for 16-bit counter). When Counter Stop Rollover is set, then interrupts are set but the counter remains at all-ones. The MMC Receive Interrupt register is a 32-bit wide register. An interrupt bit is cleared when the respective MMC counter that caused the interrupt is read. The least significant byte lane (bits[7:0]) of the respective counter must be read in order to clear the interrupt bit.

Note:

- *These registers set internally and clear when the Counter register is read.*

4.51 GMAC Register.66 (mmc_intr_tx)

This register indicates an interrupt from each transmit statistic counter.

Table 4-8 MMC Transmit Interrupt Register (mmc_intr_tx) Address:0108h

bit	R/W Permission	Initial value	Description
31:25	R	0	Reserved
24	R_SS_RC	0	This bit is set to "1" when the txvlanframes_gb counter reaches half the maximum value and also when it reaches the maximum value.
23	R_SS_RC	0	This bit is set to "1" when the txpauseframes_gb counter reaches half the maximum value and also when it reaches the maximum value.
22	R_SS_RC	0	This bit is set to "1" when the txoexcessdef counter reaches half the maximum value and also when it reaches the maximum value.
21	R_SS_RC	0	This bit is set to "1" when the txframecount counter reaches half the maximum value and also when it reaches the maximum value.
20	R_SS_RC	0	This bit is set to "1" when the txoctetcount counter reaches half the maximum value and also when it reaches the maximum value.
19	R_SS_RC	0	This bit is set to "1" when the txcarriererror counter reaches half the maximum value and also when it reaches the maximum value.
18	R_SS_RC	0	This bit is set to "1" when the txlexesscol counter reaches half the maximum value and also when it reaches the maximum value.
17	R_SS_RC	0	This bit is set to "1" when the txlatecol counter reaches half the maximum value and also when it reaches the maximum value.
16	R_SS_RC	0	This bit is set to "1" when the txdeffer counter reaches half the maximum value and also when it reaches the maximum value.
15	R_SS_RC	0	This bit is set to "1" when the txmulticol_g counter reaches half the maximum value and also when it reaches the maximum value.
14	R_SS_RC	0	This bit is set to "1" when the txsinglecol_g counter reaches half the maximum value and also when it reaches the maximum value.
13	R_SS_RC	0	This bit is set to "1" when the txunderflowerror counter reaches half the maximum value and also when it reaches the maximum value.
12	R_SS_RC	0	This bit is set to "1" when the txbroadcastframes_gb counter reaches half the maximum value and also when it reaches the maximum value.
11	R_SS_RC	0	This bit is set to "1" when the txmulticastframes_gb counter reaches half the maximum value and also when it reaches the maximum value.
10	R_SS_RC	0	This bit is set to "1" when the txunicastframes_gb counter reaches half the maximum value and also when it reaches the maximum value.
9	R_SS_RC	0	This bit is set to "1" when the tx1024tomaxoctets_gb counter reaches half the maximum value and also when it reaches the maximum value.
8	R_SS_RC	0	This bit is set to "1" when the tx512to1023octets_gb counter reaches half the maximum value and also when it reaches the maximum value.
7	R_SS_RC	0	This bit is set to "1" when the tx256to511octets_gb counter reaches half the maximum value and also when it reaches the maximum value.
6	R_SS_RC	0	This bit is set to "1" when the tx128to255maxoctects_gb counter reaches half the maximum value and also when it reaches the maximum value.

bit	R/W Permission	Initial value	Description
5	R_SS_RC	0	This bit is set to "1" when the tx65to127maxoctets_gb counter reaches half the maximum value and also when it reaches the maximum value.
4	R_SS_RC	0	This bit is set to "1" when the tx64octets_gb counter reaches half the maximum value and also when it reaches the maximum value.
3	R_SS_RC	0	This bit is set to "1" when the txmalticastframes_g counter reaches half the maximum value and also when it reaches the maximum value.
2	R_SS_RC	0	This bit is set to "1" when the txbroadcastframes_g counter reaches half the maximum value and also when it reaches the maximum value.
1	R_SS_RC	0	This bit is set to "1" when the txframecount_gb counter reaches half the maximum value and also when it reaches the maximum value.
0	R_SS_RC	0	This bit is set to "1" when the txoctetcount_gb counter reaches half the maximum value and also when it reaches the maximum value.

Each bit of the MMC Transmit Interrupt register maintains the interrupts generated when transmit statistic counters reach half their maximum values (0x8000_0000 for 32-bit counter and 0x8000 for 16-bit counter), and when they reach their maximum values (0xFFFF_FFFF for 32-bit counter and 0xFFFF for 16-bit counter). When Counter Stop Rollover is set, then interrupts are set but the counter remains at all-ones. The MMC Transmit Interrupt register is a 32-bit wide register. An interrupt bit is cleared when the respective MMC counter that caused the interrupt is read. The least significant byte lane (bits[7:0]) of the respective counter must be read in order to clear the interrupt bit.

Note:

- *These registers set internally and clear when the Counter register is read.*

4.52 GMAC Register.67 (mmc_intr_mask_rx)

This register masks an interrupt from each receive statistic counter.

Each bit of the MMC Receive Interrupt Mask register maintains the masks for the interrupts generated when receive statistic counters reach half their maximum value, and when they reach their maximum values. This register is 32 bits wide.

Table 4-9 MMC Receive Interrupt Mask Interrupt Register (mmc_intr_mask_rx) Address:010Ch

bit	R/W Permission	Initial value	Description
31:24	R	0	Reserved
23	R/W	0	This bit is set to "1" and masks the interrupt generated when the rxwatchdogerror counter reaches half the maximum value and also when it reaches the maximum value.
22	R/W	0	This bit is set to "1" and masks the interrupt generated when the rxvlanframes_gb counter reaches half the maximum value and also when it reaches the maximum value.
21	R/W	0	This bit is set to "1" and masks the interrupt generated when the rxfifooverflow_gb counter reaches half the maximum value and also when it reaches the maximum value.
20	R/W	0	This bit is set to "1" and masks the interrupt generated when the rxpauseframes counter reaches half the maximum value and also when it reaches the maximum value.
19	R/W	0	This bit is set to "1" and masks the interrupt generated when the rxoutofrangetype counter reaches half the maximum value and also when it reaches the maximum value.
18	R/W	0	This bit is set to "1" and masks the interrupt generated when the rxlengtherror counter reaches half the maximum value and also when it reaches the maximum value.
17	R/W	0	This bit is set to "1" and masks the interrupt generated when the rxunicastframes_gb counter reaches half the maximum value and also when it reaches the maximum value.
16	R/W	0	This bit is set to "1" and masks the interrupt generated when the rx1024tomaxoctets_gb counter reaches half the maximum value and also when it reaches the maximum value.
15	R/W	0	This bit is set to "1" and masks the interrupt generated when the rx512to1023octets_gb counter reaches half the maximum value and also when it reaches the maximum value.
14	R/W	0	This bit is set to "1" and masks the interrupt generated when the rx256to511octets_gb counter reaches half the maximum value and also when it reaches the maximum value.
13	R/W	0	This bit is set to "1" and masks the interrupt generated when the rx128to255maxoctects_gb counter reaches half the maximum value and also when it reaches the maximum value.
12	R/W	0	This bit is set to "1" and masks the interrupt generated when the rx65to127maxoctets_gb counter reaches half the maximum value and also when it reaches the maximum value.
11	R/W	0	This bit is set to "1" and masks the interrupt generated when the rxto64octets_gb counter reaches half the maximum value and also when it reaches the maximum value.
10	R/W	0	This bit is set to "1" and masks the interrupt generated when the rxoversize_g counter reaches half the maximum value and also when it reaches the maximum value.
9	R/W	0	This bit is set to "1" and masks the interrupt generated when the rxundersize_g counter reaches half the maximum value and also when it reaches the maximum value.
8	R/W	0	This bit is set to "1" and masks the interrupt generated when the rxjabbererror counter reaches half the maximum value and also when it reaches the maximum value.
7	R/W	0	This bit is set to "1" and masks the interrupt generated when the rxrunterror counter reaches half the maximum value and also when it reaches the maximum value.
6	R/W	0	This bit is set to "1" and masks the interrupt generated when the rxallignmenterror counter reaches half the maximum value and also when it reaches the maximum value.

bit	R/W Permission	Initial value	Description
5	R/W	0	This bit is set to "1" and masks the interrupt generated when the rxcrcerror counter reaches half the maximum value and also when it reaches the maximum value.
4	R/W	0	This bit is set to "1" and masks the interrupt generated when the rxmulticastframes_g counter reaches half the maximum value and also when it reaches the maximum value.
3	R/W	0	This bit is set to "1" and masks the interrupt generated when the rxbroadcastframes_g counter reaches half the maximum value and also when it reaches the maximum value.
2	R/W	0	This bit is set to "1" and masks the interrupt generated when the rxoctetcounter_g counter reaches half the maximum value and also when it reaches the maximum value.
1	R/W	0	This bit is set to "1" and masks the interrupt generated when the rxoctetcounter_gb counter reaches half the maximum value and also when it reaches the maximum value.
0	R/W	0	This bit is set to "1" and masks the interrupt generated when the rxframecount_gb counter reaches half the maximum value and also when it reaches the maximum value.

4.53 GMAC Register.68 (mmc_intr_mask_tx)

This register masks an interrupt from each transmit statistic counter.

Each bit of the MMC Transmit Interrupt Mask register maintains the masks for the interrupts generated when transmit statistic counters reach half their maximum value, and when they reach their maximum values. This register is 32 bits wide.

Table 4-10 MMC Transmit Interrupt Mask Register (mmc_intr_mask_tx) Address:0110h

bit	R/W Permission	Initial value	Description
31:25	R	0	Reserved
24	R/W	0	This bit is set to "1" and masks the interrupt generated when the txvlanframes_gb counter reaches half the maximum value and also when it reaches the maximum value.
23	R/W	0	This bit is set to "1" and masks the interrupt generated when the txpauseframes_gb counter reaches half the maximum value and also when it reaches the maximum value.
22	R/W	0	This bit is set to "1" and masks the interrupt generated when the txoexcessdef counter reaches half the maximum value and also when it reaches the maximum value.
21	R/W	0	This bit is set to "1" and masks the interrupt generated when the txframecount counter reaches half the maximum value and also when it reaches the maximum value.
20	R/W	0	This bit is set to "1" and masks the interrupt generated when the txoctetcount counter reaches half the maximum value and also when it reaches the maximum value.
19	R/W	0	This bit is set to "1" and masks the interrupt generated when the txcarriererror counter reaches half the maximum value and also when it reaches the maximum value.
18	R/W	0	This bit is set to "1" and masks the interrupt generated when the txlexesscol counter reaches half the maximum value and also when it reaches the maximum value.
17	R/W	0	This bit is set to "1" and masks the interrupt generated when the txlatecol counter reaches half the maximum value and also when it reaches the maximum value.
16	R/W	0	This bit is set to "1" and masks the interrupt generated when the txdeffer counter reaches half the maximum value and also when it reaches the maximum value.
15	R/W	0	This bit is set to "1" and masks the interrupt generated when the txmulticol_g counter reaches half the maximum value and also when it reaches the maximum value.
14	R/W	0	This bit is set to "1" and masks the interrupt generated when the txsinglecol_g counter reaches half the maximum value and also when it reaches the maximum value.
13	R/W	0	This bit is set to "1" and masks the interrupt generated when the txunderflowerror counter reaches half the maximum value and also when it reaches the maximum value.
12	R/W	0	This bit is set to "1" and masks the interrupt generated when the txbroadcastframes_gb counter reaches half the maximum value and also when it reaches the maximum value.
11	R/W	0	This bit is set to "1" and masks the interrupt generated when the txmulticastframes_gb counter reaches half the maximum value and also when it reaches the maximum value.
10	R/W	0	This bit is set to "1" and masks the interrupt generated when the txunicastframes_gb counter reaches half the maximum value and also when it reaches the maximum value.
9	R/W	0	This bit is set to "1" and masks the interrupt generated when the tx1024tomaxoctets_gb counter reaches half the maximum value and also when it reaches the maximum value.
8	R/W	0	This bit is set to "1" and masks the interrupt generated when the tx512to1023octets_gb counter reaches half the maximum value and also when it reaches the maximum value.
7	R/W	0	This bit is set to "1" and masks the interrupt generated when the tx256to511octets_gb counter reaches half the maximum value and also when it reaches the maximum value.

bit	R/W Permission	Initial value	Description
6	R/W	0	This bit is set to "1" and masks the interrupt generated when the tx128to255maxoctects_gb counter reaches half the maximum value and also when it reaches the maximum value.
5	R/W	0	This bit is set to "1" and masks the interrupt generated when the tx65to127maxoctets_gb counter reaches half the maximum value and also when it reaches the maximum value.
4	R/W	0	This bit is set to "1" and masks the interrupt generated when the txto64octets_gb counter reaches half the maximum value and also when it reaches the maximum value.
3	R/W	0	This bit is set to "1" and masks the interrupt generated when the txmalticastframes_g counter reaches half the maximum value and also when it reaches the maximum value.
2	R/W	0	This bit is set to "1" and masks the interrupt generated when the txbroadcastframes_g counter reaches half the maximum value and also when it reaches the maximum value.
1	R/W	0	This bit is set to "1" and masks the interrupt generated when the txframecount_gb counter reaches half the maximum value and also when it reaches the maximum value.
0	R/W	0	This bit is set to "1" and masks the interrupt generated when the txoctetcount_gb counter reaches half the maximum value and also when it reaches the maximum value.

4.54 GMAC Register.128 (mmc_ipc_intr_mask_rx)

This register masks an interrupt from each receive IPC (Checksum Offload) statistic counter.

Each bit of the MMC Receive Checksum Offload Interrupt Mask register maintains the masks for the interrupts generated when the receive IPC (Checksum Offload) statistic counters reach half their maximum value, and when they reach their maximum values. This register is 32 bits wide.

Table 4-11 MMC Receive Checksum Offload Interrupt Mask Register (mmc_ipc_intr_mask_rx) Address:0200h

bit	R/W Permission	Initial value	Description
31,30	R	0	Reserved
29	R/W	0	This bit is set to "1" and masks the interrupt generated when the rxicmp_err_octets counter reaches half the maximum value and also when it reaches the maximum value.
28	R/W	0	This bit is set to "1" and masks the interrupt generated when the rxicmp_gd_octets counter reaches half the maximum value and also when it reaches the maximum value.
27	R/W	0	This bit is set to "1" and masks the interrupt generated when the rxtcp_err_octets counter reaches half the maximum value and also when it reaches the maximum value.
26	R/W	0	This bit is set to "1" and masks the interrupt generated when the rxtcp_gd_octets counter reaches half the maximum value and also when it reaches the maximum value.
25	R/W	0	This bit is set to "1" and masks the interrupt generated when the rxudp_err_octets counter reaches half the maximum value and also when it reaches the maximum value.
24	R/W	0	This bit is set to "1" and masks the interrupt generated when the rxudp_gd_octets counter reaches half the maximum value and also when it reaches the maximum value.
23	R/W	0	This bit is set to "1" and masks the interrupt generated when the rxipv6_nopay_octets counter reaches half the maximum value and also when it reaches the maximum value.
22	R/W	0	This bit is set to "1" and masks the interrupt generated when the rxipv6_hdrerr_octets counter reaches half the maximum value and also when it reaches the maximum value.
21	R/W	0	This bit is set to "1" and masks the interrupt generated when the rxipv6_gd_octets counter reaches half the maximum value and also when it reaches the maximum value.
20	R/W	0	This bit is set to "1" and masks the interrupt generated when the rxipv4_udsbl_octets counter reaches half the maximum value and also when it reaches the maximum value.
19	R/W	0	This bit is set to "1" and masks the interrupt generated when the rxipv4_frag_octets counter reaches half the maximum value and also when it reaches the maximum value.
18	R/W	0	This bit is set to "1" and masks the interrupt generated when the rxipv4_nopay_octets counter reaches half the maximum value and also when it reaches the maximum value.
17	R/W	0	This bit is set to "1" and masks the interrupt generated when the rxipv4_hdrerr_octets counter reaches half the maximum value and also when it reaches the maximum value.
16	R/W	0	This bit is set to "1" and masks the interrupt generated when the rxipv4_gd_octets counter reaches half the maximum value and also when it reaches the maximum value.
15,14	R	0	Reserved
13	R/W	0	This bit is set to "1" and masks the interrupt generated when the rxicmp_err_frms counter reaches half the maximum value and also when it reaches the maximum value.
12	R/W	0	This bit is set to "1" and masks the interrupt generated when the rxicmp_gd_frms counter reaches half the maximum value and also when it reaches the maximum value.
11	R/W	0	This bit is set to "1" and masks the interrupt generated when the rxtcp_err_frms counter reaches half the maximum value and also when it reaches the maximum value.

bit	R/W Permission	Initial value	Description
10	R/W	0	This bit is set to "1" and masks the interrupt generated when the rxtcp_gd_frms counter reaches half the maximum value and also when it reaches the maximum value.
9	R/W	0	This bit is set to "1" and masks the interrupt generated when the rxudp_err_frms counter reaches half the maximum value and also when it reaches the maximum value.
8	R/W	0	This bit is set to "1" and masks the interrupt generated when the rxudp_gd_frms counter reaches half the maximum value and also when it reaches the maximum value.
7	R/W	0	This bit is set to "1" and masks the interrupt generated when the rxipv6_nopay_frms counter reaches half the maximum value and also when it reaches the maximum value.
6	R/W	0	This bit is set to "1" and masks the interrupt generated when the rxipv6_hdrerr_frms counter reaches half the maximum value and also when it reaches the maximum value.
5	R/W	0	This bit is set to "1" and masks the interrupt generated when the rxipv6_gd_frms counter reaches half the maximum value and also when it reaches the maximum value.
4	R/W	0	This bit is set to "1" and masks the interrupt generated when the rxipv4_udsbl_frms counter reaches half the maximum value and also when it reaches the maximum value.
3	R/W	0	This bit is set to "1" and masks the interrupt generated when the rxipv4_frag_frms counter reaches half the maximum value and also when it reaches the maximum value.
2	R/W	0	This bit is set to "1" and masks the interrupt generated when the rxipv4_nopay_frms counter reaches half the maximum value and also when it reaches the maximum value.
1	R/W	0	This bit is set to "1" and masks the interrupt generated when the rxipv4_hdrerr_frms counter reaches half the maximum value and also when it reaches the maximum value.
0	R/W	0	This bit is set to "1" and masks the interrupt generated when the rxipv4_gd_frms counter reaches half the maximum value and also when it reaches the maximum value.

4.55 GMAC Register.130 (mmc_ipc_intr_rx)

This register indicatess an interrupt from each receive IPC (Checksum Offload) statistic counter.

Each bit of the MMC Receive Checksum Offload Interrupt register maintains the interrupts generated when the receive IPC (Checksum Offload) statistic counters reach half their maximum value, and when they reach their maximum values. This register is 32 bits wide.

Table 4-12 MMC Receive Checksum Offload Interrupt Register (mmc_ipc_intr_rx) Address:0208h

bit	R/W Permission	Initial value	Description
31,30	R	0	Reserved
29	R_SS_RC	0	The bit is set to 1 when the rxicmp_err_octets counter reaches half the maximum value, and also when it reaches the maximum value.
28	R_SS_RC	0	The bit is set to 1 when the rxicmp_gd_octets counter reaches half the maximum value, and also when it reaches the maximum value.
27	R_SS_RC	0	The bit is set to 1 when the rxtcp_err_octets counter reaches half the maximum value, and also when it reaches the maximum value.
26	R_SS_RC	0	The bit is set to 1 when the rxtcp_gd_octets counter reaches half the maximum value, and also when it reaches the maximum value.
25	R_SS_RC	0	The bit is set to 1 when the rxudp_err_octets counter reaches half the maximum value, and also when it reaches the maximum value.
24	R_SS_RC	0	The bit is set to 1 when the rxudp_gd_octets counter reaches half the maximum value, and also when it reaches the maximum value.
23	R_SS_RC	0	The bit is set to 1 when the rxipv6_nopay_octets counter reaches half the maximum value, and also when it reaches the maximum value.
22	R_SS_RC	0	The bit is set to 1 when the rxipv6_hdrerr_octets counter reaches half the maximum value, and also when it reaches the maximum value.
21	R_SS_RC	0	The bit is set to 1 when the rxipv6_gd_octets counter reaches half the maximum value, and also when it reaches the maximum value.
20	R_SS_RC	0	The bit is set to 1 when the rxipv4_udsbl_octets counter reaches half the maximum value, and also when it reaches the maximum value.
19	R_SS_RC	0	The bit is set to 1 when the rxipv4_frag_octets counter reaches half the maximum value, and also when it reaches the maximum value.
18	R_SS_RC	0	The bit is set to 1 when the rxipv4_nopay_octets counter reaches half the maximum value, and also when it reaches the maximum value.
17	R_SS_RC	0	The bit is set to 1 when the rxipv4_hdrerr_octets counter reaches half the maximum value, and also when it reaches the maximum value.
16	R_SS_RC	0	The bit is set to 1 when the rxipv4_gd_octets counter reaches half the maximum value, and also when it reaches the maximum value.
15,14	R_SS_RC	0	Reserved
13	R_SS_RC	0	The bit is set to 1 when the rxicmp_err_frms counter reaches half the maximum value, and also when it reaches the maximum value.
12	R_SS_RC	0	The bit is set to 1 when the rxicmp_gd_frms counter reaches half the maximum value, and also when it reaches the maximum value.
11	R_SS_RC	0	The bit is set to 1 when the rxtcp_err_frms counter reaches half the maximum value, and also when it reaches the maximum value.

bit	R/W Permission	Initial value	Description
10	R_SS_RC	0	The bit is set to 1 when the rxtcp_gd_frms counter reaches half the maximum value, and also when it reaches the maximum value.
9	R_SS_RC	0	The bit is set to 1 when the rxudp_err_frms counter reaches half the maximum value, and also when it reaches the maximum value.
8	R_SS_RC	0	The bit is set to 1 when the rxudp_gd_frms counter reaches half the maximum value, and also when it reaches the maximum value.
7	R_SS_RC	0	The bit is set to 1 when the rxipv6_nopay_frms counter reaches half the maximum value, and also when it reaches the maximum value.
6	R_SS_RC	0	The bit is set to 1 when the rxipv6_hdrerr_frms counter reaches half the maximum value, and also when it reaches the maximum value.
5	R_SS_RC	0	The bit is set to 1 when the rxipv6_gd_frms counter reaches half the maximum value, and also when it reaches the maximum value.
4	R_SS_RC	0	The bit is set to 1 when the rxipv4_udsbl_frms counter reaches half the maximum value, and also when it reaches the maximum value.
3	R_SS_RC	0	The bit is set to 1 when the rxipv4_frag_frms counter reaches half the maximum value, and also when it reaches the maximum value.
2	R_SS_RC	0	The bit is set to 1 when the rxipv4_nopay_frms counter reaches half the maximum value, and also when it reaches the maximum value.
1	R_SS_RC	0	The bit is set to 1 when the rxipv4_hdrerr_frms counter reaches half the maximum value, and also when it reaches the maximum value.
0	R_SS_RC	0	The bit is set to 1 when the rxipv4_gd_frms counter reaches half the maximum value, and also when it reaches the maximum value.

5. Descriptors

This section explains the descriptors.

Outline of Descriptors

As described in 3.3. DMA Controller, the DMA transfers data frames received by the GMAC to the Receive Buffer in the Host memory, and transmit data frames from the Transmit Buffer in the Host memory. Descriptors that reside in the Host memory act as pointers to these buffers.
Each descriptor can point to 2 buffers, 2 byte count buffers, and 2 address pointers.
The descriptor address should be aligned to the bus width to be used.

The DMA uses enhanced descriptors.
When the Time-stamping feature and the Full IPC Offload e engine is enabled, the software needs to allocate 32-bytes (8 DWORDS) of memory for every descriptor.
When Time-stamping or Full IPC Offload engine are not enabled, the descriptor size is always 4 WORD (TDES0 to TDES3, RDES0 to REDS3)
The GMAC also needs to be configured for this change using the DMA register 0 [7].

5.1 Transmit Enhanced Descriptor

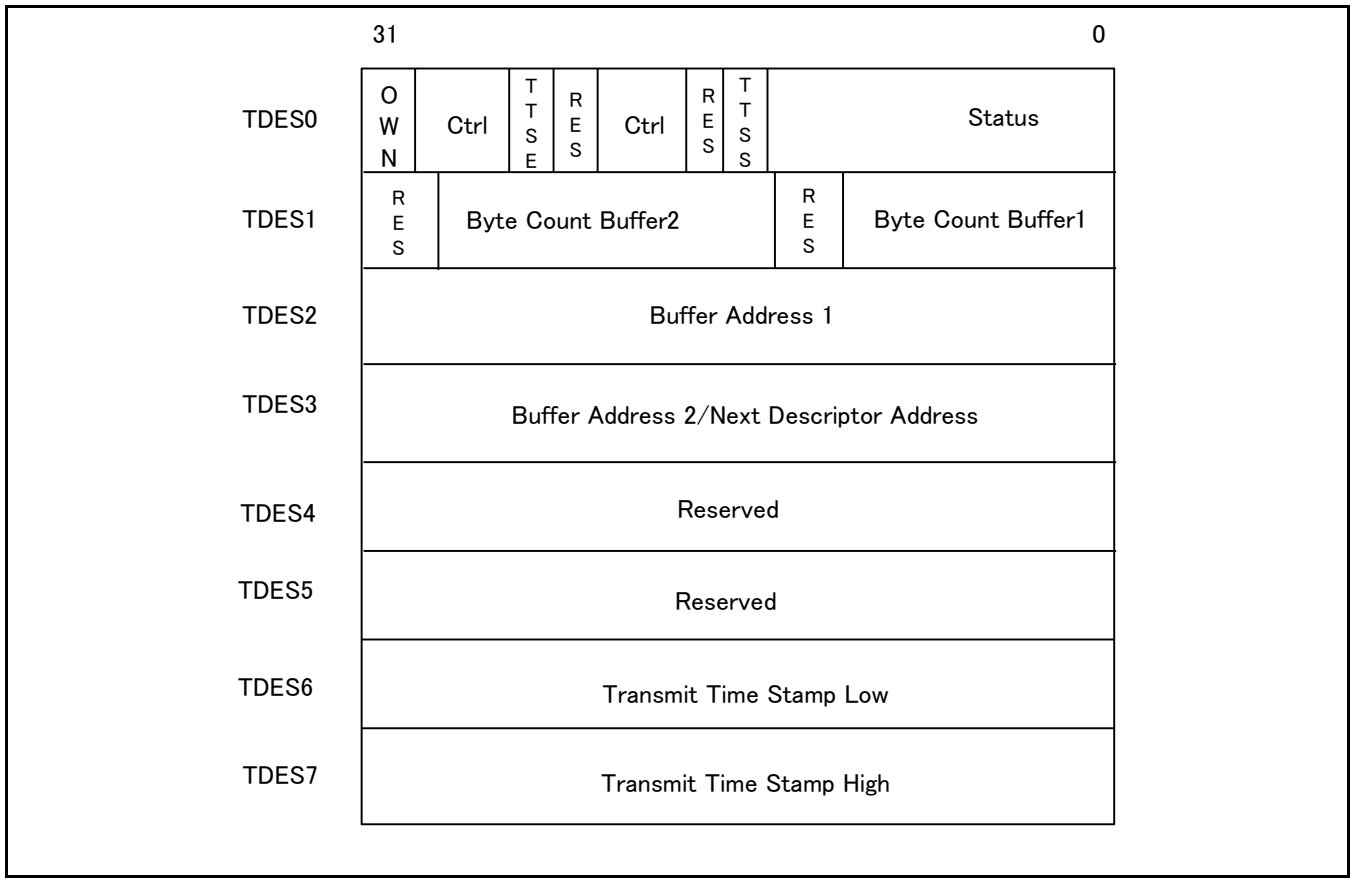
This section explains the transmit enhanced descriptor.

The transmit enhanced descriptor structure is shown in Figure 5-1. The application software must program the control bits TDES0[31:20] during descriptor initialization. When the DMA updates the descriptor, it writes back all the control bits except the OWN bit (which it clears) and updates the status bits[19:0].

Note:

- *When Time Stamp features is enabled, the software should set the DMA Bus Mode register[7], so that the DMA operates with extended descriptor size. When this control bit is reset, the TDES4-TDES7 descriptor spaces are not valid.*

Figure 5-1 Configuration Diagram of Transmit Enhanced Descriptor



5.1.1 Transmit Enhanced Descriptor 0 (TDES0)

The TDES0 consists of the control bit and status bit of the transmit frame.

TDES0

bit	31	30	29	28	27	26	25	24
Field	OWN	IC	LS	FS	DC	DP	TTSE	Reserved

bit	23	22	21	20	19	18	17	16
Field	CIC		TER	TCH	Reserved		TTSS	IHE

bit	15	14	13	12	11	10	9	8
Field	ES	JT	FF	IPE	LC	NC	LCO	EC

bit	7	6	5	4	3	2	1	0
Field	VF	CC[3:0]				ED	UF	DB

[bit31] OWN (OWN bit)

When set, this bit indicates that the descriptor is owned by the DMA. When this bit is reset, this bit indicates that the descriptor is owned by the Host. The DMA clears this bit either when it completes the frame transmission or when the buffers allocated in the descriptor are empty. The OWN bit of the First Descriptor of the frame should be set after all subsequent descriptors belonging to the same frame have been set. This avoids a possible race condition between fetching a descriptor and the driver setting the OWN bit.

[bit30] IC (Interrupt on Completion)

When this bit is set, this bit sets the Transmit Interrupt (DMA register 5[0]) after the present frame has been transmitted.

[bit29] LS (Last Segment)

When this bit is set, this bit indicates that the buffer contains the last segment of the frame.
TSB1: Transmit Buffer 1 Size or TBS2: Transmit Buffer 2 Size field in TDES1 should have a non-zero value.

[bit28] FS (First Segment)

When this bit is set, this bit indicates that the buffer contains the first segment of a frame.

[bit27] DC (Disable CRC)

When this bit is set, the Ethernet-MAC does not append a cyclic redundancy check (CRC) to the end of the transmitted frame. This is valid only when the FS (TDES0[28]) is set.

[bit26] DP (Disable Pad)

When this bit is set, the Ethernet-MAC does not automatically add padding to a frame shorter than 64 bytes. When this bit is reset, the DMA automatically adds padding and CRC field to a frame shorter than 64 bytes, despite the state of the DC (TDES0[27]) bit. This bit is valid only when the FS (TDES0[28]) is set.

[bit25] TTSE (Transmit Time Stamp Enable)

When this bit is set, this bit enables IEEE1588 hardware time stamping for the transmit frame referenced by the descriptor. This bit is valid only when the FS (TDES0[28]) is set.

[bit23, 22] CIC (Checksum Insertion Control)

These bits control the checksum calculation and insertion. Bit encodings are as shown below.

00: Checksum Insertion Disabled.
01: Only IP header checksum calculation and insertion are enabled.
10: IP header checksum and payload checksum calculation and insertion are enabled, but pseudo-header checksum is not calculated in hardware.
11: IP Header checksum and payload checksum calculation and insertion are enabled, and pseudo-header checksum is calculated in hardware.

Note:

- *In an Ethernet environment using IPv6 Authentication Header, the CIC must be 00 or 01.*

[bit21] TER (Transmit End of Ring)

When this bit is set, this bit indicates that the descriptor list reached its final descriptor. The DMA returns to the base address of the list, creating a descriptor ring.

[bit20] TCH (Second Address Chained)

When this bit is set, this bit indicates that the TDES3 address is the Next Descriptor address rather than the address of buffer 2. When this bit is set the size of buffer 2, TBS2 (TDES1[28:16]) is a "don’t care" value. When TER (TDES0[21]) is set, a value of this bit is ignored.

[bit17] TTSS (Transmit Time Stamp Status)

This bit is used as a status bit to indicate that a time stamp was captured for the described transmit frame. When this bit is set, TDES6 and TDES7 have a time stamp value captured for the transmit frame. This bit is valid only when the descriptor’s LS (TDES0[29]) is set.

[bit16] IHE (IP Header Error)

When this bit is set, this bit indicates that the GMAC transmitter detected an error in the IP datagram header. The transmitter checks the header length in the IPv4 packet against the number of header bytes received from the application and indicates an error status if there is a mismatch. For IPv6 frames, a header error is reported if the main header length is not 40 bytes. Furthermore, the Ethernet Length/Type field value for an IPv4 or IPv6 frame must match the IP header version received with the packet. For IPv4 frames, an error status is also indicated if the Header Length field has a value less than 0x5.

[bit15] ES (Error Summary)

Indicates the logical OR of the following bits:

TDES0[16] : IP Header Error
TDES0[14] : Jabber Timeout
TDES0[13] : Frame Flush
TDES0[12] : IP Payload Error
TDES0[11] : Loss of Carrier
TDES0[10] : No Carrier
TDES0[9] : Late Collision
TDES0[8] : Excessive Collision
TDES0[2] : Excessive Deferral
TDES0[1] : Underflow Error

[bit14] JT (Jabber Timeout)

When this bit is set, this bit indicates the GMAC transmitter has experienced a jabber time-out. This bit is set only when the JD bit in the GMAC register 0 is not set.

[bit13] FF (Frame Flushed)
When this bit is set, this bit indicates that the DMA/MTL flushed the frame due to a SW flush command given by the CPU.

[bit12] IPE (IP Payload Error)
When this bit is set, this bit indicates that GMAC transmitter detected an error in the TCP, UDP, or ICMP IP datagram payload. The transmitter checks the payload length received in the IPv4 or IPv6 header against the actual number of TCP, UDP, or ICMP packet bytes received from the application and issues an error status in case of a mismatch.

[bit11] LC (Loss of Carrier)
When this bit is set, this bit indicates that Loss of Carrier occurred during frame transmission (that is, the CRS signal was inactive for one or more transmit clock periods during frame transmission). This is set only for the frames transmitted without collision and when the GMAC operates in Half-Duplex Mode.

[bit10] NC (No Carrier)
When this bit is set, this bit indicates that the carrier sense signal from the PHY was not asserted during transmission.

[bit9] LCO (Late Collision)
When this bit is set, this bit indicates that frame transmission was aborted due to a collision occurring after the collision window (64 byte times including Preamble in MII Mode. Not valid if Underflow Error is set.

Note:

- *When the output of COL synchronizer is asserted after the complete frame is transmitted, it cannot be recognized as late-collision.*

[bit8] EC (Excessive Collision)
When this bit is set, this bit indicates that the transmission was aborted after 16 successive collisions while attempting to transmit the current frame. If the DR (Disable Retry) bit in the GMAC register 0 is set, this bit is set after the first collision and the transmission of the frame is aborted.

[bit7] VF (VLAN Frame)
When this bit is set, this bit indicates that the transmitted frame was a VLAN TAG Frame.

[bit6:3] CC (Collision Count)
This 4-bit counter value indicates the number of collisions occurring before the frame was transmitted. The count is not valid when the Excessive Collisions bit (TDES0[8]) is set.

[bit2] ED (Excessive Deferral)
When this bit is set, this bit indicates that the transmission has ended because of excessive deferral of over 24,288 bit times (155,680 bits times in Jumbo Frame enabled mode) if the Deferral Check (DC) bit is set high in the GMAC register 0.

[bit1] UF (Underflow Error)
When this bit is set, this bit indicates that the GMAC aborted the frame because data arrived late from the Host memory. Underflow Error indicates that the DMA encountered an empty Transmit Buffer while transmitting the frame. The transmission process enters the suspended state and sets both Transmit Underflow (DMA register 5[5]) and Transmit Interrupt (DMA register 5[0]).

[bit0] DB (Deferred Bit)
When this bit is set, this bit indicates that the GMAC defers before transmission because of the presence of carrier. This bit is valid only in Half-Duplex mode.

5.1.2 Transmit Enhanced Descriptor 1 (TDES1)

The TDES1 specifies the size of the transmit buffers 1 and 2.

TDES1

bit	31	30	29	28	27	26	25	24
Field	Reserved			TBS2[12:8]				

bit	23	22	21	20	19	18	17	16
Field	TBS2[7:0]							

bit	15	14	13	12	11	10	9	8
Field	Reserved			TBS1[12:8]				

bit	7	6	5	4	3	2	1	0
Field	TBS1[7:0]							

[bit28:16] TBS2 (Transmit Buffer 2 Size)

These bits specify the transmit data buffer 2 size in bytes. This field is not valid if TCH (TDES0[20]) is set.

[bit12:0] TBS1 (Transmit Buffer 1 Size)

These bits specify the transmit data buffer 1 size, in bytes. If this field is 0, the DMA ignores this buffer and uses transmit data buffer 2 or the next descriptor, depending on the value of TCH (TDES0[20]).

5.1.3 Transmit Enhanced Descriptor 2 (TDES2)

The TDES2 specifies the physical address of the transmit buffer 1.

TDES2								
bit	31	30	29	28	27	26	25	24
Field	B1AP[31:24]							
bit	23	22	21	20	19	18	17	16
Field	B1AP[23:16]							
bit	15	14	13	12	11	10	9	8
Field	B1AP[15:8]							
bit	7	6	5	4	3	2	1	0
Field	B1AP[7:0]							

[bit31:0] B1AP[31:0] (Buffer 1 Address Pointer)

These bits specify the physical address of Buffer 1. There is no limitation on the buffer address alignment.

5.1.4 Transmit Enhanced Descriptor 3 (TDES3)

The TDES3 specifies the physical addresses of the transmit buffer 2 and the next descriptor.

TDES3								
bit	31	30	29	28	27	26	25	24
Field	B2AP[31:24]							
bit	23	22	21	20	19	18	17	16
Field	B2AP[23:16]							
bit	15	14	13	12	11	10	9	8
Field	B2AP[15:8]							
bit	7	6	5	4	3	2	1	0
Field	B2AP[7:0]							

[bit31:0] B2AP[31:0] (Buffer 2 Address Pointer)

These bits specify the physical address of Buffer 2 when a descriptor ring structure is used. If the Second Address Chained (TDES0[20]) bit is set, this address specifies the physical memory where the Next Descriptor is present. The buffer address pointer must be aligned to the bus width only when TDES0[20] is set. (LSBs are ignored internally.).

5.1.5 Transmit Enhanced Descriptor 6 (TDES6)

The TDES6 stores the least significant 32 bits of the time stamp captured during the transmission by the DMA.

TDES6								
bit	31	30	29	28	27	26	25	24
Field	TTSL[31:24]							
bit	23	22	21	20	19	18	17	16
Field	TTSL[23:16]							
bit	15	14	13	12	11	10	9	8
Field	TTSL[15:8]							
bit	7	6	5	4	3	2	1	0
Field	TTSL[7:0]							

[bit31:0] TTSL (Transmit Frame Time Stamp Low)

This field is updated by DMA with the least significant 32 bits of the time stamp captured for the corresponding transmit frame. This field has the time stamp only if the Last Segment bit (LS) in the descriptor is set and Time stamp status (TTSS) bit is set.

5.1.6 Transmit Enhanced Descriptor 7 (TDES7)

The TDES7 stores the most significant 32 bits of the time stamp captured during the transmission by the DMA.

TDES7								
bit	31	30	29	28	27	26	25	24
Field	TTSH[31:24]							
bit	23	22	21	20	19	18	17	16
Field	TTSH[23:16]							
bit	15	14	13	12	11	10	9	8
Field	TTSH[15:8]							
bit	7	6	5	4	3	2	1	0
Field	TTSH[7:0]							

[bit31:0] TTSH (Transmit Frame Time Stamp High)

This field is updated by DMA with the most significant 32 bits of the time stamp captured for the corresponding transmit frame. This field has the time stamp only if the Last Segment bit (LS) in the descriptor is set and Time stamp status (TTSS) bit is set.

5.2 Receive Enhanced Descriptor

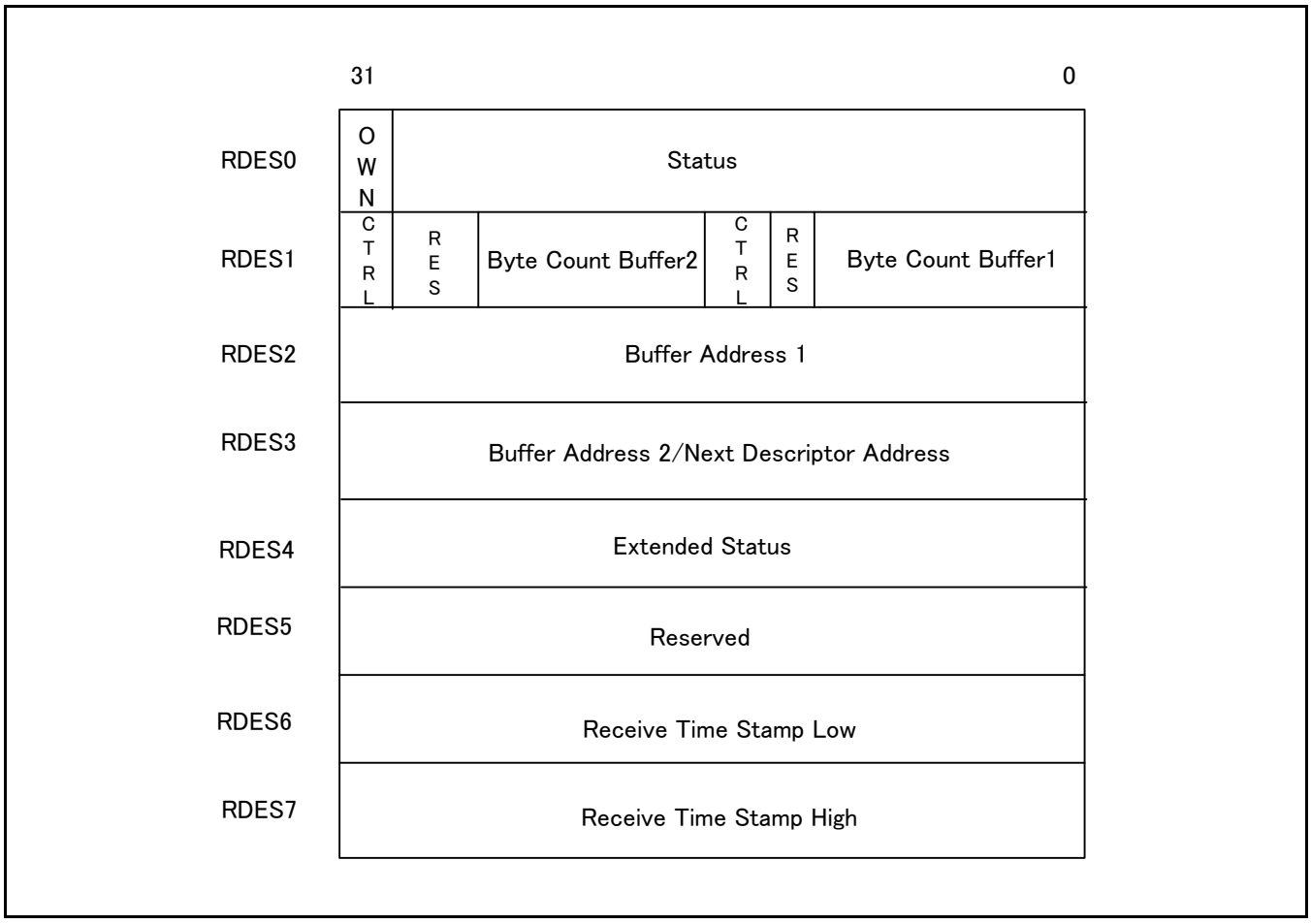
This section explains the receive descriptor.

The receive DMA requires at least two descriptors when receiving a frame. The receive state machine of the DMA always attempts to acquire an extra descriptor in anticipation of an incoming frame. (The size of the incoming frame is unknown.) Before the receive DMA closes a descriptor, it will attempt to acquire the next descriptor even if no frame are received. Figure 5-2 Configuration Diagram of Receive Enhanced Descriptor.

Note:

- *When the Time Stamp features are enabled, the software should set the DMA Bus Mode register [7], so that the DMA operates with extended descriptor size. When this control bit is reset, RDES4 to RDES7 descriptor space are not valid.*

Figure 5-2 Configuration Diagram of Receive Enhanced Descriptor



5.2.1 Receive Enhanced Descriptor 0 (RDES0)

The RDES0 consists of the control bit and status bit of the receive frame.

RDES0

bit	31	30	29	28	27	26	25	24
Field	OWN	AFM	FL[13:8]					

bit	23	22	21	20	19	18	17	16
Field	FL[7:0]							

bit	15	14	13	12	11	10	9	8
Field	ES	DE	SAF	LE	OE	VLAN	FS	LS

bit	7	6	5	4	3	2	1	0
Field	TS	LC	FT	RWT	RE	DE	CE	ESA

[bit31] OWN (OWN bit)

When this bit is set, this bit indicates that the descriptor is owned by the DMA. When this bit is reset, this bit indicates that the descriptor is owned by the Host. The DMA clears this bit either when it completes the frame reception or when the buffers that are associated with this descriptor are full.

[bit30] AFM (Destination Address Filter Fail)

When this bit is set, indicates a frame that failed in the DA Filter in the GMAC.

[bit29:16] FL[13:0] (Frame Length)

These bits indicate the byte length of the received frame that was transferred to host memory (including CRC). This field is valid when Last Descriptor (RDES0[8]) is set and either the Descriptor Error (RDES0[14]) or Overflow Error bits (RDES0[11]) are reset.

This field determines value when Last Descriptor (RDES0[8]) is set. When the Last Descriptor and Error Summary bits are not set, this field indicates the accumulated number of bytes that have been transferred for the current frame.

[bit15] ES (Error Summary)

Indicates the logical OR of the following bits:
RDES0[1] : CRC Error
RDES0[3] : Receive Error
RDES0[4] : Watchdog Timeout
RDES0[6] : Late Collision
RDES0[7] : IPC Checksum
RDES0[11]: Overflow Error
RDES0[14]: Descriptor Error
RDES4[4:3] : IP Header/Payload Error
This field is valid only when the Last Descriptor (RDES0[8]) is set.

[bit14] DE (Descriptor Error)

When this bit is set, this bit indicates a frame truncation caused by a frame that does not fit within the current descriptor buffers, and that the DMA does not own the Next Descriptor. The frame is truncated. This field is valid only when the LS (RDES0[bit8]) is set.

[bit13] SAF (Source Address Filter Fail

When this bit is set, this bit indicates that the SA field of frame failed the SA Filter in the GMAC.

[bit12] LE (Length Error)
When this bit is set, it indicates that the actual length of the frame received and Length/Type field does not match. This bit is valid only when the Frame Type (RDES0[5]) bit is reset. Length error status is not valid when CRC error is present.

[bit11] OE (Overflow Error)
When this bit is set, this bit indicates that the received frame was damaged due to buffer overflow in MTL.

[bit10] VLAN (VLAN tag)
When this bit is set, this bit indicates that the frame pointed to by this descriptor is a VLAN frame tagged by the GMAC.

[bit9] FS (First Descriptor)
When this bit is set, this bit indicates that this descriptor contains the first buffer of the frame. If the size of the first buffer is 0, the second buffer contains the beginning of the frame. If the size of the second buffer is also 0, the next Descriptor contains the beginning of the frame.

[bit8] LS (Last Descriptor)
When this bit is set, this bit indicates that the buffers pointed to by this descriptor are the last buffers of the frame.

[bit7] TS (Time Stamp)
When this bit set this bit indicates that a snapshot of the timestamp is written in descriptor words 6 (RDES6) and 7 (RDES7). This is valid only when the Last Descriptor bit (RDES0[8]) is set.

[bit6] LC (Late Collision)
When this bit is set, this bit indicates that a late collision has occurred while receiving the frame in Half-Duplex mode.

[bit5] FT (Frame Type)
When this bit is set, this bit indicates that the Receive Frame is an Ethernet-type frame (the LT field is greater than or equal to 0x0600). When this bit is reset, it indicates that the received frame is an IEEE802.3 frame. This bit is not valid for Runt frames less than 14 bytes. See Table 5-1 for details.

[bit4] RWT (Receive Watchdog Timeout)
When this bit is set, this bit indicates that the Receive Watchdog Timer has expired while receiving the current frame and the current frame is truncated after the Watchdog Timeout.

[bit3] RE (Receive Error)
When this bit is set, this bit indicates that the RX_ER signal is asserted while RX_DV is asserted during frame reception. This error also includes carrier extension error in GMII and Half-duplex mode. Error can be of less/no extension, or error (RXD ≠ 0f) during extension.

[bit2] DE (Dribble Bit Error)
When this bit is set, this bit indicates that the received frame has a non-integer multiple of bytes.

[bit1] CE (CRC Error)
When this bit is set, this bit indicates that a Cyclic Redundancy Check (CRC) Error occurred on the received frame. This field is valid only when the Last Descriptor (RDES0[8]) is set.

[bit0] ESA (Extended Status Available)
When this bit is set, indicates that the extended status is available in descriptor word 4 (RDES4). This bit is valid only when the Last Descriptor bit (RDES0[8]) is set.

5.2.2 Receive Enhanced Descriptor 1 (RDES1)

The RDES2 specifies the size of the receive buffers 1 and 2, and the control information.

RDES1

bit	31	30	29	28	27	26	25	24
Field	DIC	Reserved		RBS2[12:8]				

bit	23	22	21	20	19	18	17	16
Field	RBS2[7:0]							

bit	15	14	13	12	11	10	9	8
Field	RER	RCH	Reserved	RBS1[12:8]				

bit	7	6	5	4	3	2	1	0
Field	RBS1[7:0]							

[bit31] DIC (Disable Interrupt on Completion)

When this bit is set, this bit will prevent the setting of the RI (DMA register 5[6]) bit of the Status register for the received frame that ends in the buffer pointed to by this descriptor. This, in turn, will disable the assertion of the interrupt signal to Host due to RI for that frame.

[bit28:16] RBS2 (Receive Buffer 2 Size)

These bits indicate the size of buffer 2 in bytes. The buffer size must be a multiple of 4, even if the value of RDES3 (buffer2 address pointer) is not aligned to bus width. In the case where the buffer size is not a multiple of 4, the resulting behavior is undefined. This field is not valid if RDES1[14] is set.

[bit15] RER (Receive End of Ring)

When this bit is set, this bit indicates that the descriptor list reached its final descriptor. The DMA returns to the base address of the list, creating a Descriptor Ring.

[bit14] RCH (Second Address Chained)

When this bit is set, this bit indicates that the RDES3 address is the Next Descriptor address rather than the address of buffer 2. When this bit is set size of buffer 2, RBS2 (RDES1[28:16]) is a "don’t care" value. When RER (RDES1[15]) is set, a value of this bit is ignored.

[bit12:0] RBS1 (Receive Buffer 1 Size)

These bits indicate the size of buffer 1 in bytes. The buffer size must be a multiple of 4, 8, or 16, depending upon the bus widths, even if the value of RDES2 (buffer1 address pointer) is not aligned to bus width. When the buffer size is not a multiple of 4 the resulting behavior is undefined. If this field is 0, the DMA ignores this buffer and uses Buffer 2 or next descriptor depending on the value of RCH (RDES1[14]).

5.2.3 Receive Enhanced Descriptor 2 (RDES2)

The RDES2 specifies the physical address of the receive buffer 1.

RDES2								
bit	31	30	29	28	27	26	25	24
Field	B1AP[31:24]							
bit	23	22	21	20	19	18	17	16
Field	B1AP[23:16]							
bit	15	14	13	12	11	10	9	8
Field	B1AP[15:8]							
bit	7	6	5	4	3	2	1	0
Field	B1AP[7:0]							

[bit31:0] B1AP[31:0] (Buffer 1 Address Pointer)

These bits specify the physical address of buffer 1. There are no limitations on the buffer address alignment except for the following condition: The DMA uses the RDES2 value for its address generation of buffer 1. Note that the DMA performs a write operation to the memory in 32-bit width with the RDES2[1:0] bits as 00 during the transfer of the start of frame but the frame data is shifted as per the actual Buffer address pointer. The DMA ignores RDES2[1:0] if the address pointer is to a buffer where the middle or last part of the frame is stored.

5.2.4 Receive Enhanced Descriptor 3 (RDES3)

The RDES3 specifies the physical addresses of the receive buffer 2 and the next descriptor.

RDES3								
bit	31	30	29	28	27	26	25	24
Field	B2AP[31:24]							
bit	23	22	21	20	19	18	17	16
Field	B2AP[23:16]							
bit	15	14	13	12	11	10	9	8
Field	B2AP[15:8]							
bit	7	6	5	4	3	2	1	0
Field	B2AP[7:0]							

[bit31:0] B2AP[31:0] (Buffer 2 Address Pointer)

These bits specify the physical address of Buffer 2 when a descriptor ring structure is used. If the Second Address Chained (RDES1[14]) bit is set, this address specifies the physical address where the Next Descriptor is present.
If RDES1[14] is set, the Next Descriptor address pointer must be bus width-aligned (RDES3[1:0] = 0. LSBs are ignored internally.) However, when RDES1[14] is reset, there are no limitations on alignment of the RDES3 value, except for the following condition: The DMA uses the configured value for its buffer address generation when the RDES3 value is used to transfer the start of frame. The DMA ignores RDES3[1:0] if the RDES3 value is used for the transfer of the middle or last part of the frame.

5.2.5 Receive Enhanced Descriptor 4 (RDES4)

The RDES4 stores the enhanced status information of the receive frame.

RDES4

bit	31	30	29	28	27	26	25	24
Field	Reserved							

bit	23	22	21	20	19	18	17	16
Field	Reserved							

bit	15	14	13	12	11	10	9	8
Field	Reserved	TD	PV	PFT	MT			

bit	7	6	5	4	3	2	1	0
Field	IP6R	IP4R	IPCB	IPE	IPHE	IPT		

[bit14] TD (Timestamp Dropped)

When this bit is set, this bit indicates that the timestamp was captured for the receive frame but got dropped in the receive FIFO because of overflow.

[bit13] PV (PTP Version)

When this bit is set, indicates that the received PTP message is having the IEEE 1588 version 2 format. When reset, it has the version 1 format. This bit is valid only if the MT (message type) is not 0000.

[bit12] PFT (PTP Frame Type)

When this bit is set, this bit indicates that the PTP message over Ethernet is received. When this bit is not set and the MT (message type) is not 0000, it indicates that the PTP message over UDP-IPv4 or UDP-IPv6 is received. The information on IPv4 or IPv6 can be obtained from bits 6 and 7.

[bit11:8] MT (Message Type)

These bits are encoded to give the type of the message received.
0000 : No PTP message received or PTP packet with Reserved message type (*)
0001 : SYNC (all clock types)
0010 : Follow_Up (all clock types)
0011 : Delay_Req (all clock types)
0100 : Delay_Resp (all clock types)
0101 : Pdelay_Req (in peer-to-peer transparent clock)
0110 : Pdelay_Resp (in peer-to-peer transparent clock)
0111 : Pdelay_Resp_Follow_Up (in peer-to-peer transparent clock)
1000 : Announce
1001 : Management
1010 : Signaling
1011-1110 : Reserved
1111 : PTP packet with Reserved message type (Control Field = 0x05 to 0x0F)
* - PTP control field is 0xMN, where M=0x0 to 0x0F and N=0x0 or 0x1, and received UDP Destination Port Address is for General PTP messages.

- PTP control field is 0xMN, where M=0x1 to 0x0F and N=0x2, 0x3, or 0x4, and received UDP Destination Port Address is for Event PTP messages.

[bit7] IP6R (IPv6 Packet Received)

When this bit is set, this bit indicates that the received packet is an IPv6 packet. This bit is updated only when the IPC (Checksum Offload) of GMAC register 0[10] is set.

[bit6] IP4R (IPv4 Packet Received)

When this bit is set, this bit indicates that the received packet is an IPv4 packet. This bit is updated only when the IPC (Checksum Offload) of GMAC register 0[10] is set.

[bit5] IPCB (IP Checksum Bypassed)

When this bit is set, this bit indicates that the checksum offload engine is bypassed.

[bit4] IPE (IP Payload Error)

When this bit is set, this bit indicates that the 16-bit IP payload checksum (that is, the TCP, UDP, or ICMP checksum) that the GMAC calculated does not match the corresponding checksum field in the received segment. It is also set when the TCP, UDP, or ICMP segment length does not match the payload length value in the IP Header field.

[bit3] IPHE (IP Header Error)

When this bit is set, this bit indicates either that the 16-bit IPv4 header checksum calculated by the GMAC does not match the received checksum bytes, or that the IP datagram version is not consistent with the Ethernet Type value.

[bit2:0] IPT (IP Payload Type)

These bits indicate the type of payload encapsulated in the IP datagram processed by the Receive Checksum Offload Engine (COE). The COE also sets these bits to 000 if it does not process the IP datagram’s payload due to an IP header error or fragmented IP.

000: Unknown or did not process IP payload
001: UDP
010: TCP
011: ICMP
1xx: Reserved

5.2.6 Receive Enhanced Descriptor 6 (RDES6)

The RDES6 stores the least significant 32 bits of the time stamp captured during the reception by the DMA.

RDES6								
bit	31	30	29	28	27	26	25	24
Field	RTSL[31:24]							
bit	23	22	21	20	19	18	17	16
Field	RTSL[23:16]							
bit	15	14	13	12	11	10	9	8
Field	RTSL[15:8]							
bit	7	6	5	4	3	2	1	0
Field	RTSL[7:0]							

[bit31:0] RTSL (Receive Frame Time Stamp Low)

This field is updated by DMA with the least significant 32 bits of the time stamp captured for the corresponding receive frame. This field is updated by DMA only for the last descriptor of the receive frame that is indicated by Last Descriptor (RDES0[8]).

5.2.7 Receive Enhanced Descriptor 7 (RDES7)

The RDES7 stores the most significant 32 bits of the time stamp captured during the reception by the DMA.

RDES7

bit	31	30	29	28	27	26	25	24
Field	RTSH[31:24]							

bit	23	22	21	20	19	18	17	16
Field	RTSH[23:16]							

bit	15	14	13	12	11	10	9	8
Field	RTSH[15:8]							

bit	7	6	5	4	3	2	1	0
Field	RTSH[7:0]							

[bit31:0] RTSH (Receive Frame Time Stamp High)

This field is updated by DMA with the most significant 32 bits of the time stamp captured for the corresponding receive frame. This field is updated by DMA only for the last descriptor of the receive frame that is indicated by Last Descriptor (RDES0[8]).

6. Programming Guide

This chapter explains the programming guide for using Ethernet-MAC.

6.1 DMA Initialization/ Descriptors

1. Provide a software reset. This will reset all of the Ethernet-MAC internal registers and logic. (DMA register 0: Bus mode register - bit 0).
2. Wait for the completion of the reset process (poll the DMA register 0 : Bus Mode register[0]. This bit is cleared only after the reset operation is completed).
3. Poll the DMA register 11(AHB Status register, bit0) to confirm that all previously initiated (before software-reset) or ongoing AHB transactions are complete.
   * If the application cannot poll the AHB Status register after software reset (because of performance reasons), then you should continue with the next steps and check this register again (as mentioned in Step 12) before triggering the DMA operations.
4. If you use the MII/GMII interface, set the PS bit (GMAC register 0(MCR)[15] to1. As the initial value of this register is 0, you must write 1 to the PS bit.
5. Program the following fields to initialize the Bus Mode register by setting values in DMA register0 (Bus Mode register).
   a. MB (Mixed Burst), AAL, FB (Fixed Burst)
      The system bus of FM3 family is a multi-layer bus. The buses of CPU and the DMA of the Ethernet-MAC are independent each other.
      When the DMA of Ethernet-MAC performs the transmission with unspecified burst length(INCR), the CPU always accesses the bus. Therefore, the register settings MB=1, AAL=0, and FB=0 are recommended.
   b. PBL, RPBL, 8xPBL, USP (Burst length values and burst mode values)
   c. DSL (Descriptor length)
      When Ring Mode is selected, the setting is required.
   d. TXPR, PR, DA (Tx Rx DMA arbitration scheme)
      When the access to the system memory by the receive DMA and transmit DMA of Ethernet-MAC is conflicted, each memory access may be delayed. In this case, PR and DA can control the occupancy ratio of the receive DMA and transmit DMA.
   e. ATDS (Alternate Descriptor Size)
      When the time-stamping feature, the receive IPC, and the offload engine are selected, the setting is required.
6. Create a proper descriptor chain for transmit and receive. In addition, ensure that the receive descriptors are owned by DMA (RDES0[31] should be set). When OSF mode is used, at least two descriptors are required.
7. Make sure that your software creates three or more different transmit or receive descriptors in the ring before reusing any of the descriptors.
8. Initialize receive and transmit descriptor list address with the base address of transmit and receive descriptor (DMA register 3 - Receive Descriptor List Address register and 4 - Transmit Descriptor List Address register, respectively).
9. Program the following fields to initialize the mode of operation by setting values in DMA register6 (Operation Mode register)
   a. RSF, TSF (Receive and Transmit Store And Forward)
   b. RTC, TTC (Receive and Transmit Threshold Control)
   c. FEF, FUF (Error Frame and undersized good frame forwarding enable)
   d. OSF (Operation on Second Frame)
10. Clear the interrupt requests, by writing to those bits of the status register (interrupt bits only) which are set. For example, by writing 1 into bit16 - normal interrupt summary (NIS) will clear this bit (DMA Register5 - Status register SR[16]).

11. Enable the interrupts by programming the interrupt enable register (DMA register 7 - Interrupt enable register).
12. Read DMA register 11 (AHB Status register) to confirm that all previous AHB transactions are complete.
13. Start the Receive and Transmit DMA by setting SR (bit1) and ST (bit13) of the control register (DMA register 6 - operation mode register).

6.2 GMAC Initialization

The following GMAC Initialization operations can be performed after the DMA initialization sequence. If the GMAC Initialization is done before the DMA is set-up, then enable the GMAC receiver (last step below) only after the DMA is active. Otherwise, received frames will fill the receive FIFO and overflow.

1. Program the (GMAC register 4 - GMII Address register) for controlling the management cycles for external PHY, for example, Physical Layer Address PA (bits 15-11). Also set bit 0 (GMII Busy) for writing into PHY and reading from PHY.
2. Read the 16-bit data of GMAC register 5 (GMII Data register) from the PHY for link up, speed of operation, and mode of operation.
3. Program the appropriate fields in (GMAC register 0 - MAC configuration register) for example, Inter-frame gap while transmission, jabber disable, etc. Based on the Auto-negotiation you can set the bit11 (Duplex mode, Half-Duplex mode), bit14 (10Mpbs/100Mbps), etc. And wait 2 cycle of PHY interface clock.
4. Provide the MAC address (GMAC register 16 and GMAC register 17:MAROH, MAROL).
5. Program the Hash filter register (GMAC register 2 and GMAC register 3:MHTRH, MHTRL).
6. Program the following fields to set the appropriate filters for the incoming frames in (GMAC register 1:MFFR).
   a. Receive All
   b. Promiscuous mode
   c. Hash or Perfect Filter
   d. Unicast, Multicast, broad cast and control frames filter settings etc.
7. Program the following fields for proper flow control in (GMAC register 6:FCR).
   a. Pause time and other pause frame control bits
   b. Receive and Transmit Flow control bits
   c. Flow Control Busy/Backpressure Activate
8. Program the Interrupt Mask register bits, as required, and if applicable, for system configuration.
9. Set the bits Transmit enable (TE bit3) and Receive Enable (RE bit2) in (GMAC register 0:MCR).

6.3 Normal Receive and Transmit Operation

For normal operation, the following steps can be followed.

1. For normal transmit and receive interrupts, read the interrupt status. Then poll the descriptors, reading the status of the descriptor owned by the Host (either transmit or receive).
2. Set appropriate values for the descriptors, ensuring that transmit and receive descriptors are owned by the DMA to resume the transmission and reception of data.
3. If the descriptors were not owned by the DMA (or no descriptor is available), the DMA will go into Suspend state. The transmission or reception can be resumed by freeing the descriptors and issuing a poll demand by writing 0 into the Tx/Rx poll demand register (DMA register1 - Transmit Poll Demand register and DMA register 2 - Receive Poll Demand register)
4. The values of the current host transmitter or receiver descriptor address pointer can be read for the debug process (DMA register 18 - Current Host Transmit Descriptor register and DMA register 19 - Current Host Receive Descriptor Register).
5. The values of the transmit buffer address pointer and receive buffer address on the current host memory pointer can be read for the debug process (DMA register 20 - Current Host Transmit Buffer Address register and DMA register 21 - Current Host Receive Buffer Address register).

6.4 Stop and Start Operation

When the data transfer is required to be paused for some time then the following steps can be followed.

1. Disable the Transmit DMA (if applicable), by clearing ST (bit13) of the control register (DMA register 6:OMR).
2. Wait for any previous frame transmissions to complete. (Check by reading the DMA register 5[22:20]).
3. Flush the transmit FIFO for faster empty operation.
4. Disable the GMAC transmitter and GMAC receiver by clearing the bit3 (TE: Transmitter Enable) and bit2 (RE: Receiver Enable) in GMAC register 0:MCR).
5. After the link is up, read the PHY registers to know the latest configuration and accordingly program the GMAC registers.
6. Restart the operation by starting the transmit DMA, and then enabling the GMAC Transmitter and GMAC Receiver. You do not need to disable the receive DMA. As the GMAC Receiver is disabled, the FIFO does not get any data in the receive FIFO.

6.5 Link Down-up Sequence

When the link is down, the following steps can be followed.

1. Wait till link comes up and make sure RX and TX clocks are inputting if TX and RX clocks are stopped.
2. Software-reset to clear the DMA and GMAC Transmitter/Receiver by setting the Software Reset (SWR bit0) in DMA register 0 (Bus Mode register).
3. Follow the Initialization at "6.1. DMA Initialization/ Descriptors".

6.6 Programming Guideline for IEEE Time Stamping

Initialization Guideline for System Time Generation

The time-stamping feature can be enabled by setting bit0 of the Time Stamp control register. However, it is essential that the Time Stamp counter be initialized after this bit is set, starting time stamp operation. Do this by following these steps during Ethernet-MAC initialization:

1. Mask the Time Stamp Trigger interrupt by setting bit9 of GMAC register 15.
2. Enable the Time Stamp feature by setting bit0 of GMAC register 448: Time Stamp Control register.
3. Program the Sub-Second Increment register based on the PTP clock frequency.
4. If you are using the Fine Correction approach, program the GMAC register 454: Time Stamp Addend register and set Addend register Update (TARU: bit5) in Time Stamp Control register.
5. Poll the register until this TARU: bit5 is cleared.
6. To select the Fine Update method (if required), set TFCU: bit1 in Time Stamp Control register.
7. Program the GMAC register 452: System Time - Seconds Update register and GMAC register 453: System Time -Nanoseconds Update register with the appropriate time value.
8. Set Time Stamp Init (TSI: bit2) in Time Stamp Control register bit2.
9. The Time Stamp counter starts operation as soon as it is initialized with the value written in the Time Stamp Update register.
10. Enable the GMAC receiver and transmitter for proper time stamping.
11. Enable the Time Stamp Trigger interrupt by resetting bit9 of GMAC register 15: TMR.

Note:

- *If time stamp operation is disabled by clearing bit0 of the Time Stamp Control register, the above steps must be repeated to restart time stamp operation.*

System Time Correction

To synchronize or update the system time in one process (course correction method), perform the following steps:

1. Write the offset (positive or negative) in the Time Stamp Update registers (GMAC registers 452 and 453).
2. Set bit3 (TSU) of the Time Stamp Control register (GMAC register 448).
3. The value in the Time Stamp Update registers is added to or subtracted from the system time when the TSU bit is cleared.
4. To synchronize or update the system time to reduce system-time jitter (fine correction method), perform the following steps:
5. With the help of the algorithm explained in System Time register Module, calculate the rate by which you want to make the system time increments slower or faster.
6. Update the Time Stamp Addend register (GMAC register 454) with the new value, then set the Time Stamp Control register bit5 (TARU).
7. Wait until the system time is corrected with the new value of the Addend register to be active. You can do this by activating the Time Stamp Trigger interrupt after the system time reaches the target value.
8. Program the required target time in registers 455 and 456. Unmask the interrupt by clearing Bit 9 of GMAC register 15.
9. Set Time Stamp Control register bit4 (TITE).
10. When this trigger causes an interrupt, read GMAC register 14.
11. Reprogram the Time Stamp Addend register with the old value and set TARU(bit5) again.

6.7 Programming Guideline for Energy Efficient Ethernet

Perform the following steps during GMAC initialization:

1. Read the PHY register through the MDIO interface, check if the remote end has the EEE capability, and then negotiate the timer values.
2. Program the PHY registers through the MDIO interface. (including the RX_CLK_stoppable bit that indicates to the PHY whether to stop RX clock in LPI mode.)
3. Program the bits [25:16] (LIT: LPI LS TIMER) and bits [15:0] (TWT: LPI TW TIMER) in GMAC register 13 (LPI Timers Control register).
4. Read the link status of the PHY chip by using the MDIO interface and update the bit17 (PLS) of GMAC register 12 (LPI Control and Status register) accordingly. This update should be done whenever the link status in the PHY chip changes.
5. Set the bit16 (LPIEN: LPI Enable) of GMAC register 12 (LPI Control and Status register) to make the GMAC enter the LPI state.
   The GMAC enters the LPI mode after completing the transmission in progress and sets the bit 0 (TLPIEN: Transmit LPI Entry).

Notes:

- *If you want to make the GMAC enter the LPI state only after it completes the transmission of all queued frames in the transmit FIFO, you should set the bit 19 (LPITXA: LPI TX Automate) in GMAC register 12 (LPI Control and Status register).*
- *If you want to switch off the SYS_CLK, GMII transmit clock, or other part of the system during the LPI state, you should wait for the TLPIEN interrupt of GMAC register 12 (LPI Control and Status register) to be generated. Restore the clocks before performing the Step 6 when you want to come out of the LPI state.*

6. Reset the bit16 (LPIEN: LPI Enable) of GMAC register 12 (LPI Control and Status register) to bring the GMAC out of the LPI state.
   The GMAC waits for the time programmed in the bits [15:0] (TWT: LPI TW TIMER) before setting the TLPIEX interrupt status bit and resuming the transmission.

6.8 Transmission to Standby Mode and SYS_CLK Stop

When the WFI command is executed, the CPU transmits to the STOP mode or the Timer mode. Then, the supply of the SYS_CLK (HCLK of MCU is connected.) to Ethernet-MAC is stopped.

Therefore, when the CPU transmits to the STOP mode or Timer mode, the operation of the GMAC receiver should be stopped by the PD bit of the GMAC register 11(PMTR).

At the same time, the generation of the INI_PMT interrupt can be set by receiving the Wake-Up bucket.
The Ethernet-Mac generates the INT_PMT interrupt by receiving the Wake-Up bucket.
The CPU returns the RUN mode by the INI_PMT interrupt.

Appendixes

1. Major Changes

Spansion Publication Number: MN709-00017

Page	Section	Description
Revision 1.0		
-	-	Initial release
Revision 2.0		
6	The Target Products in This Manual	Added to TYPE5-M4.

NOTE: Please see “Document Revision History” about later revised information.

Revision History

Document Revision History

Document Title:	32-Bit Microcontroller FM4 Family Peripheral Manual Ethernet Part		
Document Number:	002-04963		
Revision	**ECN No.**	**Origin of Change**	**Description of Change**
**	-	AKIH	Migrated to Cypress and assigned document number 002-04963. No change to document contents or format.
*A	5217902	AKIH	Update to Cypress template
*B	5738079	YSAT	Adapted Cypress new logo
*C	6182746	NOSU	Perface Added the note to refer to datasheets for supported peripheral functions. Added Microcontroller support information. The Target Products in This Manual Modified part numbers in Table 1 and 2 to 8 digits description.