User's Manual

32

RZ/N1D Group, RZ/N1S Group, RZ/N1L Group

User's Manual: R-IN Engine and Ethernet Peripherals

RZ Family
RZ/N Series

All information contained in these materials, including products and product specifications, represents information on the product at the time of publication and is subject to change by Renesas Electronics Corp. without notice. Please review the latest information published by Renesas Electronics Corp. through various means, including the Renesas Electronics Corp. website (http://www.renesas.com).

Renesas Electronics
www.renesas.com

Rev.1.40 Oct, 2025

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You shall be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.
5. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
6. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The intended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below.
   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.
   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.
   Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user’s manual or other Renesas Electronics document.
7. No semiconductor product is absolutely secure. Notwithstanding any security measures or features that may be implemented in Renesas Electronics hardware or software products, Renesas Electronics shall have absolutely no liability arising out of any vulnerability or security breach, including but not limited to any unauthorized access to or use of a Renesas Electronics product or a system that uses a Renesas Electronics product. RENESAS ELECTRONICS DOES NOT WARRANT OR GUARANTEE THAT RENESAS ELECTRONICS PRODUCTS, OR ANY SYSTEMS CREATED USING RENESAS ELECTRONICS PRODUCTS WILL BE INVULNERABLE OR FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION (“Vulnerability Issues”). RENESAS ELECTRONICS DISCLAIMS ANY AND ALL RESPONSIBILITY OR LIABILITY ARISING FROM OR RELATED TO ANY VULNERABILITY ISSUES. FURTHERMORE, TO THE EXTENT PERMITTED BY APPLICABLE LAW, RENESAS ELECTRONICS DISCLAIMS ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT AND ANY RELATED OR ACCOMPANYING SOFTWARE OR HARDWARE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.
8. When using Renesas Electronics products, refer to the latest product information (data sheets, user’s manuals, application notes, “General Notes for Handling and Using Semiconductor Devices” in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
12. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
13. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
14. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.
(Note2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

(Rev.5.0-1 October 2020)

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan

www.renesas.com

Contact information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit: www.renesas.com/contact/.

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.

© 2025 Renesas Electronics Corporation. All rights reserved.

General Precautions in the Handling of Microprocessing Unit and Microcontroller Unit Products

The following usage notes are applicable to all Microprocessing unit and Microcontroller unit products from Renesas. For detailed usage notes on the products covered by this document, refer to the relevant sections of the document as well as any technical updates that have been issued for the products.

1. Precaution against Electrostatic Discharge (ESD)
   A strong electrical field, when exposed to a CMOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop the generation of static electricity as much as possible, and quickly dissipate it when it occurs. Environmental control must be adequate. When it is dry, a humidifier should be used. This is recommended to avoid using insulators that can easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors must be grounded. The operator must also be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions must be taken for printed circuit boards with mounted semiconductor devices.
2. Processing at power-on
   The state of the product is undefined at the time when power is supplied. The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the time when power is supplied. In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the time when power is supplied until the reset process is completed. In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the time when power is supplied until the power reaches the level at which resetting is specified.
3. Input of signal during power-off state
   Do not input signals or an I/O pull-up power supply while the device is powered off. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Follow the guideline for input signal during power-off state as described in your product documentation.
4. Handling of unused pins
   Handle unused pins in accordance with the directions given under handling of unused pins in the manual. The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of the LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible.
5. Clock signals
   After applying a reset, only release the reset line after the operating clock signal becomes stable. When switching the clock signal during program execution, wait until the target clock signal is stabilized. When the clock signal is generated with an external resonator or from an external oscillator during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Additionally, when switching to a clock signal produced with an external resonator or by an external oscillator while program execution is in progress, wait until the target clock signal is stable.
6. Voltage application waveform at input pin
   Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between V_{IL} (Max.) and V_{IH} (Min.) due to noise, for example, the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between V_{IL} (Max.) and V_{IH} (Min.).
7. Prohibition of access to reserved addresses
   Access to reserved addresses is prohibited. The reserved addresses are provided for possible future expansion of functions. Do not access these addresses as the correct operation of the LSI is not guaranteed.
8. Differences between products
   Before changing from one product to another, for example to a product with a different part number, confirm that the change will not lead to problems. The characteristics of a microprocessing unit or microcontroller unit products in the same group but having a different part number might differ in terms of internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

How to Use This Manual

1. Objective and Target Users

This manual was written to explain the hardware functions and electrical characteristics of this LSI to the target users, i.e. those who will be using this LSI in the design of application systems. Target users are expected to understand the fundamentals of electrical circuits, logic circuits, and microcomputers.

This manual is organized in the following items: an overview of the product, descriptions of the CPU, system control functions, and peripheral functions, electrical characteristics of the device, and usage notes.

When designing an application system that includes this LSI, take all points to note into account.
Points to note are given in their contexts and at the final part of each section, and in the section giving usage notes.

The list of revisions is a summary of major points of revision or addition for earlier versions. It does not cover all revised items. For details on the revised points, see the actual locations in the manual.

The following documents have been prepared for reference.

■ Documents related to RZ/N1

Document Name	Document Number
RZ/N1D Group, RZ/N1S Group, RZ/N1L Group DATASHEET	R01DS0323EJ****
RZ/N1D Group, RZ/N1S Group, RZ/N1L Group User's Manual: System Introduction, Multiplexing, Electrical and Mechanical Information	R01UH0750EJ****
RZ/N1D Group, RZ/N1S Group, RZ/N1L Group User's Manual: System Control and Peripheral	R01UH0751EJ****
RZ/N1D Group, RZ/N1S Group, RZ/N1L Group User's Manual: Peripherals	R01UH0752EJ****
RZ/N1D Group, RZ/N1S Group, RZ/N1L Group User's Manual: R-IN Engine and Ethernet Peripherals	R01UH0753EJ**** (this manual)
RZ/N1D Group, RZ/N1S Group, RZ/N1L Group User's Manual: PWMTimer	R01UH0913EJ****

2. Description of Registers

Each register description includes a bit chart, illustrating the arrangement of bits, and a table of bits, describing the meanings of the bit settings. The standard format and notation for bit charts and tables are described below.

X.X.X [Register Name]

(2)

Address: XXXX XXXXh

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	[Bit Field]					—	—	[Bit Field]		—	[Bit]	[Bit]	[Bit]
Value after reset	0	0	0	0	1	0	0	0	0	0	0	0	0	1	1	0

Table X.X [Register Name] Register Contents

Bit Position	Bit Name	Function	R/W
b12 to b8	[Bit Field]	[Description]	R/W
b5 to b4	[Bit Field]	[Description] 2'b00: Hi-Z 2'b01: L Output Others: Prohibited	R/W
b2	[Bit]	[Description] 1'b0: H-Z 1'b1: Output (defult)	R/W
b1	[Bit]	[Description] 1'b0: H-Z 1'b1: Output (defult)	R/W
b0	[Bit]	[Description] 1'b0: H-Z 1'b1: Output	R/W

(3)

(1)

(1) R/W: The bit or field is readable and writable.

R/(W): The bit or field is readable and writable. However, writing to this bit or field has some limitations. For details on the limitations, see the description or notes of respective registers.

R: The bit or field is readable. Writing to this bit or field has no effect.

W: The bit or field is writable. Reading to this bit or field is not guaranteed.

(2) Reserved. Make sure to use the specified value when writing to this bit or field; otherwise, the correct operation is not guaranteed.

(3) Setting prohibited. The correct operation is not guaranteed if such a setting is performed.

3. List of Abbreviations and Acronyms

Abbreviation	Full Form
AHB	Arm Advanced High-performance Bus
APB	Arm Advanced Peripheral Bus
AXI	Arm Advanced eXtensible Interface
bps	bits per second
CA7	Arm Cortex-A7 module
CM3	Arm Cortex-M3 module
CRC	Cyclic Redundancy Check
DMA	Direct Memory Access
DMAC	Direct Memory Access Controller
Hi-Z	High Impedance
HSR	High-availability Seamless Redundancy
HW-RTOS	Hard Ware Real Time OS
I/O	Input/Output
LSB	Least Significant Bit
MSB	Most Significant Bit
NC	Non-Connect
NoC	Network-on-Chip
PLL	Phase Locked Loop
PWM	Pulse Width Modulation
UART	Universal Asynchronous Receiver/Transmitter
OTP	One Time Programmable ROM
PTP	Precision Time Protocol
PRP	Parallel Redundancy Protocol
SoC	System On Chip

4. Description of the Access Size

Access size:

8 bits = Byte

16 bits = Halfword

32 bits = Word

EtherCAT is registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany.
Sercos is a registered trademark of Sercos International e.V.
ETHERNET Powerlink is the registered trademark of ETHERNET Powerlink Standardization Group (EPSG).
Arm is a registered trademark of Arm Limited (or its subsidiaries) in the EU and/or elsewhere. All rights reserved.
All trademarks and registered trademarks are the property of their respective owners.

Table of Contents

Section 1 R-IN Engine Outline 31
1.1 Overview 31

Section 2 Hardware Real Time OS (HW-RTOS) 33
2.1 Overview 33

Section 3 Gigabit Ethernet MAC (HW-RTOS GMAC) 34
3.1 Overview 34
3.2 Signal Interface 35
3.3 Register Map 36
3.3.1 HW-RTOS HWFC (Hardware Function Call) Register Map 36
3.3.2 HW-RTOS GMAC (Gigabit Ethernet MAC) Register Map 36
3.4 Register Description 37
3.4.1 HW-RTOS HWFC (Hardware Function Call) Register Description 37
3.4.1.1 C0TYPE — Hardware Function Type Register 37
3.4.1.2 C0STAT — Hardware Function Status Register 37
3.4.1.3 SYSC — Hardware Function System Call Register 38
3.4.1.4 R4 — Hardware Function Argument Register 4 39
3.4.1.5 R5 — Hardware Function Argument Register 5 39
3.4.1.6 R6 — Hardware Function Argument Register 6 40
3.4.1.7 R7 — Hardware Function Argument Register 7 40
3.4.1.8 CMD — Hardware Function Command Register 41
3.4.1.9 R0 — Hardware Function Return Value Register 0 41
3.4.1.10 R1 — Hardware Function Return Value Register 1 42
3.4.2 HW-RTOS GMAC (Gigabit Ethernet MAC) Register Description 43
3.4.2.1 GMAC_TXID — TX ID Register 43
3.4.2.2 GMAC_TXRESULT — TX Result Register 43
3.4.2.3 GMAC_MODE — Mode Register 44
3.4.2.4 GMAC_RXMODE — RX Mode Register 45
3.4.2.5 GMAC_TXMODE — TX Mode Register 47
3.4.2.6 GMAC_RESET — Reset Register 48
3.4.2.7 GMAC_PAUSE[m] —Pause Packet Data Register [m] (m = 1 to 5) 48
3.4.2.8 GMAC_FLWCTL — RX Flow Control Register 49
3.4.2.9 GMAC_PAUSPKT — Pause Packet Register 49
3.4.2.10 GMAC_MIIM — MIIM Register 50
3.4.2.11 GMAC_ADR[m]A — MAC Address Register [m]A (m = 1 to 16) 51
3.4.2.12 GMAC_ADR[m]B — MAC Address Register [m]B (m = 1 to 16) 51
3.4.2.13 GMAC_RXFIFO — RX FIFO Status Register 52
3.4.2.14 GMAC_TXFIFO — TX FIFO Status Register 53
3.4.2.15 GMAC_ACC — TCP/IPACC Register 54
3.4.2.16 GMAC_RXMAC_ENA — RX MAC Enable Register 55
3.4.2.17 BUFID — Receive Buffer Information Register 56
3.5 Operation 57
3.5.1 Initialization 57
3.5.2 Hardware Functions (HWF) 59

3.5.2.1 Processing Flow 60
3.5.2.2 Buffer Allocator 61
3.5.2.3 MAC DMA Controller 67
3.5.2.4 Buffer RAM DMA Controller 78
3.5.3 Interrupts 82
3.5.4 Ethernet Frame Transmission Function 84
3.5.4.1 Get a Transmit Buffer 84
3.5.4.2 Create Transmit Data 84
3.5.4.3 Create Transmit Descriptors 90
3.5.4.4 Transmission Start 91
3.5.4.5 Completing Transmission 91
3.5.5 Ethernet Frame Reception Function 92
3.5.5.1 Enable Reception MAC 92
3.5.5.2 Enable Reception DMA 92
3.5.5.3 Receive a Frame and Get a Buffer 92
3.5.5.4 Get Receive Buffer Information 92
3.5.5.5 Format of Receive Data 92
3.5.6 TCPIP Accelerator Function 101
3.6 Usage Notes 103
3.6.1 Giga Bit Ethernet MAC (HW-RTOS GMAC) Control Register 103
3.6.1.1 MIIM Register (GMAC_MIIM) 103
3.6.1.2 TX ID Register (GMAC_TXID) 103
3.6.1.3 TX Result Register (GMAC_TXRESULT) 103
3.6.1.4 TX Mode Register (GMAC_TXMODE) 104
3.6.1.5 Reset Register (GMAC_RESET) 104
3.6.1.6 Pause Packet Data Register [m] (GMAC_PAUSE[m]) 104
3.6.2 Hardware Function Call Register 105
3.6.3 Management TAG Control 105
3.6.4 Notice 106
3.6.4.1 Padding Added to the MAC Header in the Transmit Frame 106
3.6.4.2 Erroneous Judgment about Checksum Validation at Specific Frame Reception 106
3.6.4.3 Error of Receive Frame Information at Rx FIFO Overflow 106
3.6.4.4 Error of Rx Frame Information at Reception of the Frame More than 64 bytes with Padding 112
3.6.4.5 About Jumbo Frame 113

Section 4 Advanced 5port Switch (A5PSW) 114
4.1 Overview 114
4.2 Signal Interface 118
4.3 Register Map 119
4.4 Register Description 125
4.4.1 REVISION — Switch Core Version 125
4.4.2 SCRATCH — Scratch Register 125
4.4.3 PORT_ENA — Port Enable Register 126
4.4.4 UCAST_DEFAULT_MASK — Unicast Default Mask Register 127
4.4.5 VLAN_VERIFY — Verify VLAN Domain 127
4.4.6 BCAST_DEFAULT_MASK — Broadcast Default Mask Register 128

4.4.7 MCAST_DEFAULT_MASK — Multicast Default Mask Register 128
4.4.8 INPUT_LEARN_BLOCK — Input Learning Block Register 129
4.4.9 MGMT_CONFIG — Management Configuration Register 130
4.4.10 MODE_CONFIG — Mode Configuration Register 131
4.4.11 VLAN_IN_MODE — VLAN Input Manipulation Mode Register 132
4.4.12 VLAN_OUT_MODE — VLAN Output Manipulation Mode Register 133
4.4.13 VLAN_IN_MODE_ENA — VLAN Input Mode Enable Register 134
4.4.14 VLAN_TAG_ID — VLAN Tag ID Register 134
4.4.15 BCAST_STORM_LIMIT — Broadcast Storm Protection Register 135
4.4.16 MCAST_STORM_LIMIT — Multicast Storm Protection Register 135
4.4.17 MIRROR_CONTROL — Port Mirroring Configuration Register 136
4.4.18 MIRROR_EG_MAP — Port Mirroring Egress Port Definition 137
4.4.19 MIRROR_ING_MAP — Port Mirroring Ingress Port Definition 137
4.4.20 MIRROR_ISRC_0 — Ingress Source MAC Address for Mirror Filtering 0 138
4.4.21 MIRROR_ISRC_1 — Ingress Source MAC Address for Mirror Filtering 1 138
4.4.22 MIRROR_IDST_0 — Ingress Destination MAC Address for Mirror Filtering 0 139
4.4.23 MIRROR_IDST_1 — Ingress Destination MAC Address for Mirror Filtering 1 139
4.4.24 MIRROR_ESRC_0 — Egress Source MAC Address for Mirror Filtering 0 140
4.4.25 MIRROR_ESRC_1 — Egress Source MAC Address for Mirror Filtering 1 140
4.4.26 MIRROR_EDST_0 — Egress Destination MAC Address for Mirror Filtering 0 141
4.4.27 MIRROR_EDST_1 — Egress Destination MAC Address for Mirror Filtering 1 141
4.4.28 MIRROR_CNT — Mirror Filtering Count Value Register 142
4.4.29 QMGR_ST_MINCELLS — Output Queue Minimum Memory Statistics Register 142
4.4.30 QMGR_RED_MIN4 — RED Minimum Threshold Register 143
4.4.31 QMGR_RED_MAX4 — RED Maximum Threshold Register 143
4.4.32 QMGR_RED_CONFIG — RED Configuration Register 144
4.4.33 IMC_STATUS — Input Memory Controller Status Register 145
4.4.34 IMC_ERR_FULL — Input Port Memory Full and Truncation Indicator 146
4.4.35 IMC_ERR_IFACE — Input Port Memory Error Indicator 147
4.4.36 IMC_ERR_QOFLOW — Output Port Queue Overflow Indicator 148
4.4.37 IMC_CONFIG — Input Memory Controller Configuration Register 148
4.4.38 GPARSER[n] — [n]th Parser of 1st Block (n = 0..3) 149
4.4.39 GARITH[n] — Snoop Configuration for Arithmetic [n]th Stage of 1st Block (n = 0..3) 151
4.4.40 GPARSER[n] — [n−4]th Parser of 2nd Block (n = 4..7) 152
4.4.41 GARITH[n] — Snoop Configuration for Arithmetic [n−4]th Stage of 2nd Block (n = 4..7) .154
4.4.42 VLAN_PRIORITY[n] — VLAN Priority Register [n] (n = 0..4) 155
4.4.43 IP_PRIORITY[n] — IP Priority Register [n] (n = 0..4) 156
4.4.44 PRIORITY_CFG[n] — Priority Configuration Register [n] (n = 0..4) 157
4.4.45 PRIORITY_TYPE1 — Priority Type Register 1 158
4.4.46 PRIORITY_TYPE2 — Priority Type Register 2 159
4.4.47 MGMT_ADDR0_lo — Lower MAC Address for Bridge Protocol Frame 160
4.4.48 MGMT_ADDR0_hi — Higher MAC Address for Bridge Protocol Frame 160
4.4.49 SYSTEM_TAGINFO[n] — One VLAN ID Field for VLAN Input Manipulation (n = 0..4) 161

4.4.50 AUTH_PORT[n] — PORT[n] Authentication Control and Configuration (n = 0..4) 162
4.4.51 VLAN_RES_TABLE[n] — 32 VLAN Domain Entries (n = 0..31) 164
4.4.52 TOTAL_DISC — Discarded Frame Total Number Register 165
4.4.53 TOTAL_BYT_DISC — Discarded Frame Total Bytes Register 165
4.4.54 TOTAL_FRM — Processed Frame Total Number Register 166
4.4.55 TOTAL_BYT_FRM — Processed Frame Total Bytes Register 166
4.4.56 ODISC[n] — PORT[n] Discarded Outgoing Frame Count Register (n = 0..4) 167
4.4.57 IDISC_VLAN[n] — PORT[n] Discarded Incoming VLAN Tagged Frame Count Register (n = 0..4) 167
4.4.58 IDISC_UNTAGGED[n] — PORT[n] Discarded Incoming VLAN Untagged Frame Count Register (n = 0..4) 168
4.4.59 IDISC_BLOCKED[n] — PORT[n] Discarded Incoming Blocked Frame Count Register (n = 0..4) 168
4.4.60 IMC_QLEVEL_P[n] — PORT[n] Queued Frame Count Register (n = 0..4) 169
4.4.61 LK_CTRL — Learning/Lookup Function Global Configuration Register 170
4.4.62 LK_STATUS — Status Bits and Table Overflow Counter 172
4.4.63 LK_ADDR_CTRL — Address Table Transaction Control and Read/Write Address 173
4.4.64 LK_DATA_LO — Lower 32-Bit Data of Lookup Memory Entry 175
4.4.65 LK_DATA_HI — Higher 26-Bit Data of Lookup Memory Entry 175
4.4.66 LK_LEARNCOUNT — Learned Address Count Register 176
4.4.67 LK_AGETIME — Period of the Aging Timer 177
4.4.68 MGMT_TAG_CONFIG — Management Tag Configuration Register 178
4.4.69 PEERDELAY[n] — Peer Delay Value for Port [n] (n = 0..3) 179
4.4.70 PORT[n]_CTRL — PORT[n] Timestamp Control/Status (n = 0..3) 180
4.4.71 PORT[n]_TIME — PORT[n] Memorized Transmit Timestamp (n = 0..3) 181
4.4.72 INT_CONFIG — Interrupt Enable Configuration Register 182
4.4.73 INT_STAT_ACK — Interrupt Status/ACK Register 184
4.4.74 MDIO_CFG_STATUS — MDIO Configuration and Status Register 186
4.4.75 MDIO_COMMAND — MDIO PHY Command Register 187
4.4.76 MDIO_DATA — MDIO Data Register 187
4.4.77 REV_P[n] — PORT[n] MAC Core Revision (n = 0..4) 188
4.4.78 COMMAND_CONFIG_P[n] — Port[n] Command Configuration Register (n = 0..4) 189
4.4.79 MAC_ADDR_0_P[n] — Port[n] MAC Address Register 0 (n = 0..3) 191
4.4.80 MAC_ADDR_1_P[n] — Port[n] MAC Address Register 1 (n = 0..3) 191
4.4.81 FRM_LENGTH_P[n] — PORT[n] Maximum Frame Length Register (n = 0..4) 192
4.4.82 PAUSE_QUANT_P[n] — PORT[n] MAC Pause Quanta (n = 0..4) 192
4.4.83 PTPClockIdentity1_P[n] — PORT[n] PTP Clock Identity1 Register (n = 0..3) 193
4.4.84 PTPClockIdentity2_P[n] — PORT[n] PTP Clock Identity2 Register (n = 0..3) 193
4.4.85 PTPAutoResponse_P[n] — PORT[n] PTP Auto Response Register (n = 0..3) 194
4.4.86 STATUS_P[n] — PORT[n] Port Status Register (n = 0..4) 195
4.4.87 TX_IPG_LENGTH_P[n] — PORT[n] Transmit IPG Length Register (n = 0..4) 196
4.4.88 EEE_CTL_STAT_P[n] — PORT[n] MAC EEE Functions Control and Status (n = 0..3) 197
4.4.89 EEE_IDLE_TIME_P[n] — PORT[n] EEE Idle Time Register (n = 0..3) 199

4.4.90 EEE_TWSYS_TIME_P[n] — PORT[n] EEE Wake Up Time Register (n = 0..3) 199
4.4.91 IDLE_SLOPE_P[n] — PORT[n] MAC Traffic Shaper Bandwidth Control (n = 0..4) 200
4.4.92 aFramesTransmittedOK_P[n] — PORT[n] MAC Transmitted Valid Frame Count Register (n = 0..4) 200
4.4.93 aFramesReceivedOK_P[n] —PORT[n] MAC Received Valid Frame Count Register (n = 0..4) 201
4.4.94 aFrameCheckSequenceErrors_P[n] —PORT[n] MAC FCS Error Frame Count Register (n = 0..4) 201
4.4.95 aAlignmentErrors_P[n] —PORT[n] MAC Alignment Error Frame Count Register (n = 0..4) 202
4.4.96 aOctetsTransmittedOK_P[n] —PORT[n] MAC Transmitted Valid Frame Octets Register (n = 0..4) 202
4.4.97 aOctetsReceivedOK_P[n] —PORT[n] MAC Received Valid Frame Octets Register (n = 0..4) 203
4.4.98 aTxPAUSEMACCtrlFrames_P[n] —PORT[n] MAC Transmitted Pause Frame Count Register (n = 0..4) 203
4.4.99 aRxPAUSEMACCtrlFrames_P[n] — PORT[n] MAC Received Pause Frame Count Register (n = 0..4) 204
4.4.100 ifInErrors_P[n] — PORT[n] MAC Input Error Count Register (n = 0..4) 204
4.4.101 ifOutErrors_P[n] — PORT[n] MAC Output Error Count Register (n = 0..4) 205
4.4.102 ifInUcastPkts_P[n] — PORT[n] MAC Received Unicast Frame Count Register (n = 0..4) 205
4.4.103 ifInMulticastPkts_P[n] — PORT[n] MAC Received Multicast Frame Count Register (n = 0..4) 206
4.4.104 ifInBroadcastPkts_P[n] — PORT[n] MAC Received Broadcast Frame Count Register (n = 0..4) 206
4.4.105 ifOutDiscards_P[n] — PORT[n] MAC Discarded Outbound Frame Count Register (n = 0..4) 207
4.4.106 ifOutUcastPkts_P[n] — PORT[n] MAC Transmitted Unicast Frame Count Register (n = 0..4) 207
4.4.107 ifOutMulticastPkts_P[n] — PORT[n] MAC Transmitted Multicast Frame Count Register (n = 0..4) 208
4.4.108 ifOutBroadcastPkts_P[n] — PORT[n] MAC Transmitted Broadcast Frame Count Register (n = 0..4) 208
4.4.109 etherStatsDropEvents_P[n] — PORT[n] MAC Dropped Frame Count Register (n = 0..4) 209
4.4.110 etherStatsOctets_P[n] — PORT[n] MAC All Frame Octets Register (n = 0..4) 209
4.4.111 etherStatsPkts_P[n] — PORT[n] MAC All Frame Count Register (n = 0..4) 210
4.4.112 etherStatsUndersizePkts_P[n] — PORT[n] MAC Too Short Frame Count Register (n = 0..4) 210
4.4.113 etherStatsOversizePkts_P[n] — PORT[n] MAC Too Long Frame Count Register (n = 0..4) 211

4.4.114 etherStatsPkts64Octets_P[n] — PORT[n] MAC 64 Octets Frame Count Register (n = 0..4) 211
4.4.115 etherStatsPkts65to127Octets_P[n] — PORT[n] MAC 65 to 127 Octets Frame Count Register (n = 0..4) 212
4.4.116 etherStatsPkts128to255Octets_P[n] — PORT[n] MAC 128 to 255 Octets Frame Count Register (n = 0..4) 212
4.4.117 etherStatsPkts256to511Octets_P[n] — PORT[n] MAC 256 to 511 Octets Frame Count Register (n = 0..4) 213
4.4.118 etherStatsPkts512to1023Octets_P[n] — PORT[n] MAC 512 to 1023 Octets Frame Count Register (n = 0..4) 213
4.4.119 etherStatsPkts1024to1518Octets_P[n] — PORT[n] MAC 1024 to 1519 Octets Frame Count Register (n = 0..4) 214
4.4.120 etherStatsPkts1519toXOctets_P[n] — PORT[n] MAC Over 1519 Octets Frame Count Register (n = 0..4) 214
4.4.121 etherStatsJabbers_P[n] — PORT[n] MAC Jabbers Frame Count Register (n = 0..4) 215
4.4.122 etherStatsFragments_P[n] — PORT[n] MAC Fragment Frame Count Register (n = 0..4) 215
4.4.123 VLANReceivedOK_P[n] — PORT[n] MAC Received VLAN Tagged Frame Count Register (n = 0..4) 216
4.4.124 VLANTransmittedOK_P[n] — PORT[n] MAC Transmitted VLAN Tagged Frame Count Register (n = 0..4) 216
4.4.125 FramesRetransmitted_P[n] — PORT[n] MAC Retransmitted Frame Count Register (n = 0..4) 217
4.4.126 STATS_HIWORD_P[n] — PORT[n] MAC Statistics Counter High Word Register (n = 0..4) 218
4.4.127 STATS_CTRL_P[n] — PORT[n] MAC Statistics Control Register (n = 0..4) 219
4.4.128 STATS_CLEAR_VALUELO_P[n] — PORT[n] MAC Statistics Clear Value Lower Register (n = 0..4) 220
4.4.129 STATS_CLEAR_VALUEHI_P[n] — PORT[n] MAC Statistics Clear Value Higher Register (n = 0..4) 220
4.4.130 aDeferred_P[n] — PORT[n] MAC Deferred Count Register (n = 0..3) 221
4.4.131 aMultipleCollisions_P[n] — PORT[n] MAC Multiple Collision Count Register (n = 0..3) 221
4.4.132 aSingleCollisions_P[n] — PORT[n] MAC Single Collision Count Register (n = 0..3) 222
4.4.133 aLateCollisions_P[n] — PORT[n] MAC Late Collision Count Register (n = 0..3) 222
4.4.134 aExcessiveCollisions_P[n] — PORT[n] MAC Excessive Collision Count Register (n = 0..3) 223
4.4.135 aCarrierSenseErrors_P[n] — PORT[n] MAC Carrier Sense Error Count Register (n = 0..3) 223
4.4.136 DLR_CONTROL — DLR Control Register 224
4.4.137 DLR_STATUS — DLR Status Register 225
4.4.138 DLR_ETH_TYP — DLR Ethernet Type Register 226
4.4.139 DLR_IRQ_CONTROL — DLR Interrupt Control Register 227
4.4.140 DLR_IRQ_STAT_ACK — DLR Interrupt Status/ACK Register 228

4.4.141 DLR_LOC_MAClo — DLR Local MAC Address Low Register 230
4.4.142 DLR_LOC_MAChi — DLR Local MAC Address High Register 230
4.4.143 DLR_SUPR_MAClo — DLR Supervisor MAC Address Low Register 231
4.4.144 DLR_SUPR_MAChi — DLR Supervisor MAC Address High Register 231
4.4.145 DLR_STATE_VLAN — DLR Ring Status/VLAN Register 232
4.4.146 DLR_BEC_TMOUT — DLR Beacon Timeout Register 232
4.4.147 DLR_BEC_INTRVL — DLR Beacon Interval Register 233
4.4.148 DLR_SUPR_IPADR — DLR Supervisor IP Address Register 233
4.4.149 DLR_ETH_STYP_VER — DLR Sub Type/Protocol Version Register 234
4.4.150 DLR_INV_TMOUT — DLR Beacon Timeout Timer Register 234
4.4.151 DLR_SEQ_ID — DLR Sequence ID Register 235
4.4.152 DLR_DSTlo — DLR Beacon Destination Address Low Register 235
4.4.153 DLR_DSThi — DLR Beacon Destination Address High Register 236
4.4.154 DLR_RX_STAT0 — DLR Received Frame Statistic Register 0 236
4.4.155 DLR_RX_ERR_STAT0 — DLR Received Frame Error Statistic Register 0 237
4.4.156 DLR_TX_STAT0 — DLR Transmitted Frame Statistic Register 0 237
4.4.157 DLR_RX_STAT1 — DLR Received Frame Statistic Register 1 238
4.4.158 DLR_RX_ERR_STAT1 — DLR Received Frame Error Statistic Register 1 238
4.4.159 DLR_TX_STAT1 — DLR Transmitted Frame Statistic Register 1 239
4.4.160 PRP_CONFIG — PRP Configuration Register 240
4.4.161 PRP_GROUP — PRP Port Group Register 242
4.4.162 PRP_SUFFIX — PRP RCT Suffix 243
4.4.163 PRP_LANID — PRP LAN Identifier 243
4.4.164 DUP_W — PRP Max Duplicate Detection Window Size 244
4.4.165 PRP_AGETIME — PRP Aging Time Define Register 244
4.4.166 PRP_IRQ_CONTROL — PRP Interrupt Control Register 245
4.4.167 PRP_IRQ_STAT_ACK — PRP Interrupt Status/ACK Register 246
4.4.168 RM_ADDR_CTRL — PRP History Memory Transactions Control Register 247
4.4.169 RM_DATA — PRP Memory Data Register 248
4.4.170 RM_STATUS — PRP Memory Controller Status Indication 248
4.4.171 TxSeqTooLate — PRP Frame Transmission Retrieval of Failed Sequence 249
4.4.172 CntErrWrongLanA — PRP Wrong ID LAN-A Count Register 249
4.4.173 CntErrWrongLanB — PRP Wrong ID LAN-B Count Register 250
4.4.174 CntDupLanA — PRP Duplicate LAN-A Count Register 250
4.4.175 CntDupLanB — PRP Duplicate LAN-B Count Register 251
4.4.176 CntOutOfSeqLowA — PRP Sequence Error Low LAN-A Count Register 251
4.4.177 CntOutOfSeqLowB — PRP Sequence Error Low LAN-B Count Register 252
4.4.178 CntOutOfSeqA — PRP Sequence Error LAN-A Count Register 252
4.4.179 CntOutOfSeqB — PRP Sequence Error LAN-B Count Register 253
4.4.180 CntAcceptA — PRP Valid Frame LAN-A Count Register 253
4.4.181 CntAcceptB — PRP Valid Frame LAN-B Count Register 254
4.4.182 CntMissing — PRP Drop history Adjustment Count 254
4.4.183 HUB_CONFIG — HUB Configuration Register 255

4.4.184 HUB_GROUP — HUB Port Group Register 256
4.4.185 HUB_DEFPORT — HUB Default Port Selection Register 257
4.4.186 HUB_TRIGGER_IMMEDIATE — HUB Transmission Trigger Immediate Register 258
4.4.187 HUB_TRIGGER_AT — HUB Transmission Trigger At Register 259
4.4.188 HUB_TTIME — HUB Transmission Time Define Register 259
4.4.189 HUB_IRQ_CONTROL — HUB Interrupt Control Register 260
4.4.190 HUB_IRQ_STAT_ACK — HUB Interrupt Status/ACK Register 261
4.4.191 HUB_STATUS — HUB Status Register 262
4.4.192 HUB_OPORT_STATUS — HUB Output Port Status Register 263
4.4.193 RXMATCH_CONFIG[n] — RX Pattern Match Configuration for PORT[n] (n = 0..4) 263
4.4.194 PATTERN_CTRL[n] — RX Pattern Match Function Control for Pattern[n] (n = 0..7) 264
4.4.195 PTN_IRQ_CONTROL — RX Pattern Match Interrupt Control Register 266
4.4.196 PTN_IRQ_STAT_ACK — RX Pattern Match Interrupt Status/ACK Register 267
4.4.197 PATTERN_SEL — RX Pattern Number Selection Register 267
4.4.198 PTRN_CMP_30 — Pattern Compare Value Bytes 3 .. 0 268
4.4.199 PTRN_CMP_74 — Pattern Compare Value Bytes 7 .. 4 268
4.4.200 PTRN_CMP_118 — Pattern Compare Value Bytes 11 .. 8 269
4.4.201 PTRN_MSK_30 — Pattern Mask for Bytes 3 .. 0 269
4.4.202 PTRN_MSK_74 — Pattern Mask for Bytes 7 .. 4 270
4.4.203 PTRN_MSK_118 — Pattern Mask for Bytes 11 .. 8 270
4.4.204 TDMA_CONFIG — TDMA Configuration Register 271
4.4.205 TDMA_PORTS — TDMA Scheduling Enable Register 272
4.4.206 TDMA_START — TDMA Start Time Set Register 272
4.4.207 TDMA_MODULO — TDMA System Timer Modulo 273
4.4.208 TDMA_CYCLE — TDMA Periodic Cycle Set Register 273
4.4.209 TDMA_T1 — TDMA 1st Time Offset 274
4.4.210 TDMA_T2 — TDMA 2nd Time Offset 274
4.4.211 TDMA_T3 — TDMA 3rd Time Offset 275
4.4.212 QUEUES_TS — TDMA 1st Slot Transmit Enable 276
4.4.213 QUEUES_T1 — TDMA 2nd Slot Transmit Enable 276
4.4.214 QUEUES_T2 — TDMA 3rd Slot Transmit Enable 277
4.4.215 QUEUES_T3 — TDMA Last Slot Transmit Enable 277
4.4.216 QUEUES_START — TDMA First Cycle Transmit Enable 278
4.4.217 TIME_LOAD_NEXT — TDMA Calculated Next Loading Time 278
4.4.218 TDMA_IRQ_CONTROL — TDMA Interrupt Control Register 279
4.4.219 TDMA_IRQ_STAT_ACK — TDMA Interrupt Status/ACK Register 280
4.5 Operation 281
4.5.1 Ethernet Frame Format Overview 281
4.5.1.1 Overview 281
4.5.1.2 MAC Address Overview 283
4.5.1.3 VLAN Tag Overview 283
4.5.1.4 Pause Frames 284
4.5.2 IP Frame Format 285
4.5.2.1 Definitions 285

4.5.2.2 Ethernet Types 285
4.5.2.3 IPv4 Datagram Format 285
4.5.2.4 IPv6 Datagram Format 286
4.5.3 Switch Functional Description 288
4.5.3.1 Overview 288
4.5.3.2 VLAN Input Processing Function 288
4.5.3.3 VLAN Output Processing Function 290
4.5.3.4 Frame Snooping 291
4.5.3.5 Frame Classification and Priority Resolution 296
4.5.3.6 Input Port Selection 299
4.5.3.7 Layer 2 Look Up Engine 299
4.5.3.8 Layer 2 Look Up Engine Operational Description 301
4.5.3.9 Frame Forwarding Tasks 304
4.5.3.10 Broadcast Storm Protection 311
4.5.3.11 Multicast Storm Protection 311
4.5.3.12 Output Frame Queuing 311
4.5.3.13 Output Port Congestion Management 312
4.5.4 Timestamping Functions (TSM) 314
4.5.4.1 Overview 314
4.5.4.2 IEEE 1588 Message Header Structure 314
4.5.4.3 Timestamp Processing 315
4.5.4.4 Transparent Clock Support 315
4.5.4.5 Using One Step update at the Management Port 317
4.5.4.6 Automatic Response to Peer Delay Request Messages 318
4.5.5 Management Port Specific Frame Tagging 320
4.5.5.1 Tagged Frame Format 320
4.5.5.2 Byte Order 321
4.5.5.3 Transmit Processing (Switch to CPU) 321
4.5.5.4 Receive Processing (CPU to Switch) 321
4.5.5.5 Module Configuration 323
4.5.6 MAC EEE Support 324
4.5.6.1 Overview 324
4.5.6.2 PHY Interface Encodings for EEE 325
4.5.7 MAC Transmit Rate Control 326
4.5.7.1 General 326
4.5.7.2 Configuration Settings 326
4.5.8 Port Based Access Control (802.1X) 327
4.5.8.1 General 327
4.5.8.2 Terms & Definitions 327
4.5.8.3 Functions 327
4.5.8.4 Forwarding Decision 328
4.5.8.5 Usage aspects 329
4.5.9 Interrupts 330
4.5.10 PHY Management Interface (MDIO Master) 331
4.5.10.1 Overview 331
4.5.10.2 MDIO Frame Format 331
4.5.10.3 Turnaround Signalling 332
4.5.10.4 MDIO Control Registers 332
4.5.10.5 Transaction Types 332
4.5.11 DLR Extension Module 333
4.5.11.1 Overview 333

4.5.11.2 Beacon Frame Format 334
4.5.11.3 Ring Node Functional Description 337
4.5.12 Parallel Redundancy Protocol (PRP) 340
4.5.12.1 Overview 340
4.5.12.2 A5PSW Function Summary 341
4.5.12.3 Switch Forwarding Behavior 343
4.5.12.4 MAC Address Learning Enhancements 345
4.5.13 Integrated HUB Module 347
4.5.13.1 Overview 347
4.5.13.2 Functional Description 348
4.5.13.3 Hub Specific Forwarding Rules 349
4.5.13.4 HUB Group Clocking 350
4.5.13.5 PHY Requirement 350
4.5.13.6 Usage Information 350
4.5.14 Receive Pattern Matcher 351
4.5.14.1 Functional Description 351
4.5.14.2 Usage Information 351
4.5.15 TDMA Operation 352
4.5.15.1 Overview 352
4.5.15.2 Output Queue Scheduling 352
4.5.15.3 Scheduling Example 352
4.5.15.4 Startup Sequence 354
4.5.15.5 Usage Information 354
4.5.16 Initializing A5PSW 355
4.6 Usage notes 356
4.6.1 Restriction 356
4.6.2 References 358

Section 5 EtherCAT Slave Controller 359
5.1 Overview 359
5.1.1 Block Diagram 361
5.1.2 Features 362
5.2 Signal Interface 363
5.3 Register Map 364
5.4 Register Description 367
5.4.1 TYPE — Type Register 367
5.4.2 REVISION — Revision Register 367
5.4.3 BUILD — Build Register 368
5.4.4 FMMU_NUM — FMMUs Supported Register 368
5.4.5 SYNC_MANAGER — SyncManagers Supported Register 368
5.4.6 RAM_SIZE — RAM Size Register 369
5.4.7 PORT_DESC — Port Descriptor Register 370
5.4.8 FEATURE — ESC Features Supported Register 371
5.4.9 STATION_ADR — Configured Station Address Register 372
5.4.10 STATION_ALIAS — Configured Station Alias Register 372
5.4.11 WR_REG_ENABLE — Write Register Enable Register 373

5.4.12 WR_REG_PROTECT — Write Register Protection Register 373
5.4.13 ESC_WR_ENABLE — ESC Write Enable Register 374
5.4.14 ESC_WR_PROTECT — ESC Write Protection Register 374
5.4.15 ESC_RESET_ECAT — ESC Reset EtherCAT Register 375
5.4.16 ESC_RESET_PDI — ESC Reset PDI Register 376
5.4.17 ESC_DL_CONTROL — ESC DL Control Register 377
5.4.18 PHYSICAL_RW_OFFSET — Physical Read/Write Offset Register 379
5.4.19 ESC_DL_STATUS — ESC DL Status Register 380
5.4.20 AL_CONTROL — AL Control Register 382
5.4.21 AL_STATUS — AL Status Register 383
5.4.22 AL_STATUS_CODE — AL Status Code Register 383
5.4.23 RUN_LED_OVERRIDE — RUN LED Override Register 384
5.4.24 ERR_LED_OVERRIDE — ERR LED Override Register 385
5.4.25 PDI_CONTROL — PDI Control Register 386
5.4.26 ESC_CONFIG — ESC Configuration Register 387
5.4.27 PDI_CONFIG — PDI Configuration Register 388
5.4.28 SYNC_LATCH_CONFIG — SYNC/LATCH PDI Configuration Register 389
5.4.29 EXT_PDI_CONFIG — Extended PDI Configuration Register 390
5.4.30 ECAT_EVENT_MASK — EtherCAT Event Mask Register 390
5.4.31 AL_EVENT_MASK — AL Event Mask Register 391
5.4.32 ECAT_EVENT_REQ — EtherCAT Event Request Register 392
5.4.33 AL_EVENT_REQ — AL Event Request Register 394
5.4.34 RX_ERR_COUNT[n] — Rx Error Counter [n] Register (n = 0..2) 396
5.4.35 FWD_RX_ERR_COUNT[n] — Forwarded Rx Error Counter [n] Register (n = 0..2) 397
5.4.36 ECAT_PROC_ERR_COUNT — EtherCAT Processing Unit Error Counter Register 397
5.4.37 PDI_ERR_COUNT — PDI Error Counter Register 398
5.4.38 LOST_LINK_COUNT[n] — Lost Link Counter [n] Register (n = 0..2) 398
5.4.39 WD_DIVIDE — Watchdog Divider Register 399
5.4.40 WDT_PDI — Watchdog Time PDI Register 399
5.4.41 WDT_DATA — Watchdog Time Process Data Register 400
5.4.42 WDS_DATA — Watchdog Status Process Data Register 400
5.4.43 WDC_DATA — Watchdog Counter Process Data Register 401
5.4.44 WDC_PDI — Watchdog Counter PDI Register 401
5.4.45 EEP_CONF — EEPROM Configuration Register 402
5.4.46 EEP_STATE — EEPROM PDI Access State Register 402
5.4.47 EEP_CONT_STAT — EEPROM Control/Status Register 403
5.4.48 EEP_ADR — EEPROM Address Register 404
5.4.49 EEP_DATA — EEPROM Data Register 405
5.4.50 MII_CONT_STAT — MII Management Control/Status Register 406
5.4.51 PHY_ADR — PHY Address Register 407
5.4.52 PHY_REG_ADR — PHY Register Address Register 408
5.4.53 PHY_DATA — PHY Data Register 408
5.4.54 MII_ECAT_ACS_STAT — MII Management EtherCAT Access State Register 409

5.4.55 MII_PDI_ACS_STAT — MII Management PDI Access State Register 410
5.4.56 PHY_STATUS[n] — PHY Port Status [n] Register (n = 0..2) 411
5.4.57 FMMU[n]_L_START_ADR — FMMU Logical Start Address [n] Register (n = 0..7).....412
5.4.58 FMMU[n]_LEN — FMMU Length [n] Register (n = 0..7).....412
5.4.59 FMMU[n]_L_START_BIT — FMMU Logical Start Bit [n] Register (n = 0..7).....413
5.4.60 FMMU[n]_L_STOP_BIT — FMMU Logical Stop bit [n] Register (n = 0..7) 413
5.4.61 FMMU[n]_P_START_ADR — FMMU Physical Start Address [n] Register (n = 0..7) 414
5.4.62 FMMU[n]_P_START_BIT — FMMU Physical Start Bit [n] Register (n = 0..7) 414
5.4.63 FMMU[n]_TYPE — FMMU Type [n] Register (n = 0..7) 415
5.4.64 FMMU[n]_ACT — FMMU Activate [n] Register (n = 0..7).....415
5.4.65 SM[n]_P_START_ADR — SyncManager Physical Start Address [n] Register (n = 0..7) .416
5.4.66 SM[n]_LEN — SyncManager Length [n] Register (n = 0..7).....416
5.4.67 SM[n]_CONTROL — SyncManager Control [n] Register (n = 0..7) 417
5.4.68 SM[n]_STATUS — SyncManager Status [n] Register (n = 0..7) 418
5.4.69 SM[n]_ACT — SyncManager Activate [n] Register (n = 0..7).....419
5.4.70 SM[n]_PDI_CONT — SyncManager PDI Control [n] Register (n = 0..7).....420
5.4.71 DC_RCV_TIME_PORT0 — Receive Times Port0 Register 421
5.4.72 DC_RCV_TIME_PORT1 — Receive Times Port1 Register 422
5.4.73 DC_RCV_TIME_PORT2 — Receive Times Port2 Register 422
5.4.74 DC_SYS_TIME — System Time Register 423
5.4.75 DC_RCV_TIME_UNIT — Receive Time EtherCAT Processing Unit Register.....424
5.4.76 DC_SYS_TIME_OFFSET — System Time Offset Register 425
5.4.77 DC_SYS_TIME_DELAY — System Time Delay Register.....426
5.4.78 DC_SYS_TIME_DIFF — System Time Difference Register 426
5.4.79 DC_SPEED_COUNT_START — Speed Counter Start Register 427
5.4.80 DC_SPEED_COUNT_DIFF — Speed Counter Diff Register 427
5.4.81 DC_SYS_TIME_DIFF_FIL_DEPTH — System Time Difference Filter Depth Register 428
5.4.82 DC_SPEED_COUNT_FIL_DEPTH — Speed Counter Filter Depth Register 428
5.4.83 DC_CYC_CONT — Cyclic Unit Control Register 429
5.4.84 DC_ACT — Activation Register 430
5.4.85 DC_PULSE_LEN — Pulse Length of SyncSignals Register 431
5.4.86 DC_ACT_STAT — Activation Status Register 432
5.4.87 DC_SYNC0_STAT — SYNC0 Status Register 433
5.4.88 DC_SYNC1_STAT — SYNC1 Status Register 433
5.4.89 DC_CYC_START_TIME — Start Time Cyclic Operation/Next SYNC0 Pulse Register 434
5.4.90 DC_NEXT_SYNC1_PULSE — Next SYNC1 Pulse Register.....435
5.4.91 DC_SYNC0_CYC_TIME — SYNC0 Cycle Time Register 436
5.4.92 DC_SYNC1_CYC_TIME — SYNC1 Cycle Time Register 436
5.4.93 DC_LATCH0_CONT — Latch0 Control Register 437
5.4.94 DC_LATCH1_CONT — Latch1 Control Register 437
5.4.95 DC_LATCH0_STAT — Latch0 Status Register.....438
5.4.96 DC_LATCH1_STAT — Latch1 Status Register.....439
5.4.97 DC_LATCH0_TIME_POS — Latch0 Time Positive Edge Register.....440

5.4.98 DC_LATCH0_TIME_NEG — Latch0 Time Negative Edge Register 441
5.4.99 DC_LATCH1_TIME_POS — Latch1 Time Positive Edge Register 442
5.4.100 DC_LATCH1_TIME_NEG — Latch1 Time Negative Edge Register 443
5.4.101 DC_ECAT_CNG_EV_TIME — Buffer Change Event Time Register 444
5.4.102 DC_PDI_START_EV_TIME — PDI Buffer Start Event Time Register 444
5.4.103 DC_PDI_CNG_EV_TIME — PDI Buffer Change Event Time Register 445
5.4.104 PRODUCT_ID — PRODUCT ID Register 446
5.4.105 VENDOR_ID — Vendor ID Register 447
5.4.106 USER_RAM — User RAM 448
5.4.107 DATA_RAM — Process Data RAM 449
5.5 Operation 450
5.5.1 Initializing 450
5.6 Usage notes 452
5.6.1 Restriction 452
5.6.2 Reset Circuit 452

Section 6 Ethernet MAC 10/100/1000 (GMAC) 454
6.1 Overview 454
6.2 Signal Interface 456
6.3 Register Map 457
6.3.1 GMAC1 Register Map 457
6.3.2 GMAC2 Register Map 460
6.4 Register Description 463
6.4.1 MAC_Configuration — MAC Configuration Register 463
6.4.2 MAC_Frame_Filter — MAC Frame Filter Register 466
6.4.3 GMII_Address — GMII Address Register 468
6.4.4 GMII_Data — GMII Data Register 469
6.4.5 Flow_Control — Flow Control Register 470
6.4.6 VLAN_Tag — VLAN Tag Register 472
6.4.7 Version — Version Register 473
6.4.8 Debug — Debug Register 474
6.4.9 Remote_Wake_Up_Frame_Filter — Remote Wake-Up Frame Filter Register 475
6.4.10 PMT_Control_Status — PMT Control and Status Register 476
6.4.11 LPI_Control_Status — LPI Control and Status Register 478
6.4.12 LPI_Timers_Control — LPI Timers Control Register 479
6.4.13 Interrupt_Status — Interrupt Status Register 480
6.4.14 Interrupt_Mask — Interrupt Mask Register 481
6.4.15 MAC_Address0_High — MAC Address 0 High Register 482
6.4.16 MAC_Address0_Low — MAC Address 0 Low Register 482
6.4.17 MAC_Address[n]_High — MAC Address [n] High Register (n = 1..17) 483
6.4.18 MAC_Address[n]_Low — MAC Address [n] Low Register (n = 1..17) 484
6.4.19 WDog_Timeout — Watchdog Timeout Register 485

6.4.20 MMC_Control — MMC Control Register 486
6.4.21 MMC_Receive_Interrupt — MMC Receive Interrupt Register.....487
6.4.22 MMC_Transmit_Interrupt — MMC Transmit Interrupt Register.....489
6.4.23 MMC_Receive_Interrupt_Mask — MMC Receive Interrupt Mask Register 491
6.4.24 MMC_Transmit_Interrupt_Mask — MMC Transmit Interrupt Mask Register 493
6.4.25 Tx_Octet_Count_Good_Bad — Transmit Octet Count for Good and Bad Frames.....495
6.4.26 Tx_Frame_Count_Good_Bad — Transmit Frame Count for Good and Bad Frames.....495
6.4.27 Tx_Broadcast_Frames_Good — Transmit Frame Count for Good Broadcast Frames 496
6.4.28 Tx_Multicast_Frames_Good — Transmit Frame Count for Good Multicast Frames 496
6.4.29 Tx_64Octets_Frames_Good_Bad — Transmit Octet Count for Good and Bad 64 Byte Frames 497
6.4.30 Tx_65To127Octets_Frames_Good_Bad — Transmit Octet Count for Good and Bad 65 to 127 Bytes Frames 497
6.4.31 Tx_128To255Octets_Frames_Good_Bad — Transmit Octet Count for Good and Bad 128 to 255 Bytes Frames 498
6.4.32 Tx_256To511Octets_Frames_Good_Bad — Transmit Octet Count for Good and Bad 256 to 511 Bytes Frames 498
6.4.33 Tx_512To1023Octets_Frames_Good_Bad — Transmit Octet Count for Good and Bad 512 to 1023 Bytes Frames 499
6.4.34 Tx_1024ToMaxOctets_Frames_Good_Bad — Transmit Octet Count for Good and Bad 1024 to Maxsize Bytes Frames 499
6.4.35 Tx_Unicast_Frames_Good_Bad — Transmit Frame Count for Good and Bad Unicast Frames 500
6.4.36 Tx_Multicast_Frames_Good_Bad — Transmit Frame Count for Good and Bad Multicast Frames.....500
6.4.37 Tx_Broadcast_Frames_Good_Bad — Transmit Frame Count for Good and Bad Broadcast Frames.....501
6.4.38 Tx_Underflow_Error_Frames — Transmit Frame Count for Underflow Error Frames.....501
6.4.39 Tx_Single_Collision_Good_Frames — Transmit Frame Count for Frames Transmitted after Single Collision 502
6.4.40 Tx_Multiple_Collision_Good_Frames — Transmit Frame Count for Frames Transmitted after Multiple Collision.....502
6.4.41 Tx_Deferred_Frames — Transmit Frame Count for Deferred Frames 503
6.4.42 Tx_Late_Collision_Frames — Transmit Frame Count for Late Collision Error Frames 503
6.4.43 Tx_Excessive_Collision_Frames — Transmit Frame Count for Excessive Collision Error Frames 504
6.4.44 Tx_Carrier_Error_Frames — Transmit Frame Count for Carrier Sense Error Frames 504
6.4.45 Tx_Octet_Count_Good — Transmit Octet Count for Good Frames.....505
6.4.46 Tx_Frame_Count_Good — Transmit Frame Count for Good Frames 505
6.4.47 Tx_Excessive_Deferral_Error — Transmit Frame Count for Excessive Deferral Error Frames 506
6.4.48 Tx_Pause_Frames — Transmit Frame Count for Good PAUSE Frames 506
6.4.49 Tx_VLAN_Frames_Good — Transmit Frame Count for Good VLAN Frames.....507

6.4.50 Tx_OSize_Frames_Good — Transmit Frame Count for Good Oversize Frames.....507
6.4.51 Rx_Frames_Count_Good_Bad — Receive Frame Count for Good and Bad Frames.....508
6.4.52 Rx_Octet_Count_Good_Bad — Receive Octet Count for Good and Bad Frames 508
6.4.53 Rx_Octet_Count_Good — Receive Octet Count for Good Frames 509
6.4.54 Rx_Broadcast_Frames_Good — Receive Frame Count for Good Broadcast Frames.....509
6.4.55 Rx_Multicast_Frames_Good — Receive Frame Count for Good Multicast Frames.....510
6.4.56 Rx_CRC_Error_Frames — Receive Frame Count for CRC Error Frames 510
6.4.57 Rx_Alignment_Error_Frames — Receive Frame Count for Alignment Error Frames.....511
6.4.58 Rx_Runt_Error_Frames — Receive Frame Count for Runt Error Frames.....511
6.4.59 Rx_Jabber_Error_Frames — Receive Frame Count for Jabber Error Frames.....512
6.4.60 Rx_Undersize_Frames_Good — Receive Frame Count for Undersize Frames.....512
6.4.61 Rx_Oversize_Frames_Good — Receive Frame Count for Oversize Frames.....513
6.4.62 Rx_64Octets_Frames_Good_Bad — Receive Frame Count for Good and Bad 64 Byte Frames 513
6.4.63 Rx_65To127Octets_Frames_Good_Bad — Receive Frame Count for Good and Bad 65 to 127 Bytes Frames 514
6.4.64 Rx_128To255Octets_Frames_Good_Bad — Receive Frame Count for Good and Bad 128 to 255 Bytes Frames 514
6.4.65 Rx_256To511Octets_Frames_Good_Bad — Receive Frame Count for Good and Bad 256 to 511 Bytes Frames 515
6.4.66 Rx_512To1023Octets_Frames_Good_Bad — Receive Frame Count for Good and Bad 512 to 1,023 Bytes Frames 515
6.4.67 Rx_1024ToMaxOctets_Frames_Good_Bad — Receive Frame Count for Good and Bad 1,024 to Maxsize Bytes Frames 516
6.4.68 Rx_Unicast_Frames_Good — Receive Frame Count for Good Unicast Frames 516
6.4.69 Rx_Length_Error_Frames — Receive Frame Count for Length Error Frames.....517
6.4.70 Rx_Out_Of_Range_Type_Frames — Receive Frame Count for Out of Range Frames ..517
6.4.71 Rx_Pause_Frames — Receive Frame Count for PAUSE Frames.....518
6.4.72 Rx_FIFO_Overflow_Frames — Receive Frame Count for FIFO Overflow Frames 518
6.4.73 Rx_VLAN_Frames_Good_Bad — Receive Frame Count for Good and Bad VLAN Frames 519
6.4.74 Rx_Watchdog_Error_Frames — Receive Frame Count for Watchdog Error Frames 519
6.4.75 Rx_Receive_Error_Frames — Receive Frame Count for Receive Error Frames.....520
6.4.76 Rx_Control_Frames_Good — Receive Frame Count for Good Control Frames 520
6.4.77 Hash_Table_Reg[n] — Hash Table Register [n] (n = 0..7).....521
6.4.78 VLAN_Hash_Table_Reg — VLAN Hash Table Register 521
6.4.79 Timestamp_Control — Timestamp Control Register 522
6.4.80 Sub_Second_Increment — Sub-Second Increment Register 524
6.4.81 System_Time_Seconds — System Time - Seconds Register 524
6.4.82 System_Time_Nanoseconds — System Time - Nanoseconds Register 525
6.4.83 System_Time_Seconds_Update — System Time - Seconds Update Register 525
6.4.84 System_Time_Nanoseconds_Update — System Time - Nanoseconds Update Register.....526

6.4.85 Timestamp_Addend — Timestamp Addend Register 526
6.4.86 Target_Time_Seconds — Target Time Seconds Register 527
6.4.87 Target_Time_Nanoseconds — Target Time Nanoseconds Register 528
6.4.88 Timestamp_Status — Timestamp Status Register 529
6.4.89 PPS_Control — PPS Control Register 531
6.4.90 Auxiliary_Timestamp_Nanoseconds — Auxiliary Timestamp – Nanoseconds Register...533
6.4.91 Auxiliary_Timestamp_Seconds — Auxiliary Timestamp - Seconds Register.....533
6.4.92 PPS0_Interval — PPS0 Interval Register 534
6.4.93 PPS0_Width — PPS0 Width Register 534
6.4.94 PPS1_Target_Time_Seconds — PPS1 Target Time Seconds Register 535
6.4.95 PPS1_Target_Time_Nanoseconds — PPS1 Target Time Nanoseconds Register 536
6.4.96 PPS1_Interval — PPS1 Interval Register 537
6.4.97 PPS1_Width — PPS1 Width Register 537
6.4.98 Bus_Mode — Bus Mode Register 538
6.4.99 Transmit_Poll_Demand — Transmit Poll Demand Register 540
6.4.100 Receive_Poll_Demand — Receive Poll Demand Register 540
6.4.101 Receive_Descriptor_List_Address — Receive Descriptor List Address Register 541
6.4.102 Transmit_Descriptor_List_Address — Transmit Descriptor List Address Register 541
6.4.103 Status — Status Register 542
6.4.104 Operation_Mode — Operation Mode Register 545
6.4.105 Interrupt_Enable — Interrupt Enable Register 548
6.4.106 Missed_Frame_And_Buffer_Overflow_Counter — Missed Frame and Buffer Overflow Counter Register 550
6.4.107 Receive_Interrupt_Watchdog_Timer — Receive Interrupt Watchdog Timer Register.....551
6.4.108 AXI_Bus_Mode — AXI Bus Mode Register 552
6.4.109 AXI_Status — AXI Status Register 553
6.4.110 Current_Host_Transmit_Descriptor — Current Host Transmit Descriptor Register 554
6.4.111 Current_Host_Receive_Descriptor — Current Host Receive Descriptor Register 554
6.4.112 Current_Host_Transmit_Buffer_Address — Current Host Transmit Buffer Address Register 555
6.4.113 Current_Host_Receive_Buffer_Address — Current Host Receive Buffer Address Register 555
6.4.114 HW_Feature — HW Feature Register 556
6.5 Operation 557
6.5.1 Initializing 557
6.5.1.1 Initializing Operation 558
6.5.2 Initializing DMA 559
6.5.3 Initializing GMAC 561
6.5.4 Performing Normal Receive and Transmit Operation 562
6.5.5 Stopping and Starting Transmission 562
6.5.6 Programming Guidelines for GMII Link Transitions 563
6.5.6.1 Transmit and Receive Clocks are Running when the Link is Down 563
6.5.6.2 Transmit and Receive Clocks are Stopped when the Link is Down 563

6.5.7 Programming Guidelines for IEEE 1588 Timestamping 564
6.5.7.1 Initialization Guideline for System Time Generation....564
6.5.7.2 System Time Correction 565
6.5.8 System Time Register Module....566
6.5.9 Programming Guidelines for Energy Efficient Ethernet 569
6.5.9.1 Entering and Exiting the Tx LPI Mode 569
6.5.9.2 Gating Off the CSR Clock in the LPI Mode 570
6.5.10 Programming Guidelines for Flexible Pulse Per Second (PPS) Output (GMAC1 only) 571
6.5.10.1 Generating Single Pulse on PPS....571
6.5.10.2 Generating a Pulse Train on PPS....572
6.5.10.3 Generating an Interrupt without Affecting the PPS....573
6.5.11 Enhanced Descriptors....574
6.5.11.1 Transmit Descriptor....575
6.5.11.2 Receive Descriptor....580
6.6 Usage Notes 585
6.6.1 Configurations 585
6.6.2 Restriction 585
6.6.2.1 Ethernet MAC 585
6.6.3 MAC Management Counters 586
6.6.3.1 Address Assignments 586
6.6.4 MMC Registers 588
6.6.4.1 MMC Control Register (MMC_Control)....588
6.6.4.2 MMC Receive Interrupt Register (MMC_Receive_Interrupt)....588
6.6.4.3 MMC Transmit Interrupt Register (MMC_Transmit_Interrupt)....588
6.6.4.4 MMC Receive Interrupt Mask Register (MMC_Receive_Interrupt_Mask) 588
6.6.4.5 MMC Transmit Interrupt Mask Register (MMC_Transmit_Interrupt_Mask) 588
6.6.5 Power Management Block 589
6.6.5.1 PMT Block Description....590
6.6.6 DMA Register....593
6.6.6.1 Bus Mode Register (Bus_Mode)....593
6.6.6.2 Transmit Poll Demand Register (Transmit_Poll_Demand)....593
6.6.6.3 Receive Poll Demand Register (Receive_Poll_Demand)....593
6.6.6.4 Receive Descriptor List Address Register (Receive_Descriptor_List_Address) 593
6.6.6.5 Transmit Descriptor List Address Register (Transmit_Descriptor_List_Address) ..593
6.6.6.6 Status Register (Status)....594
6.6.6.7 Operation Mode Register (Operation_Mode) 594
6.6.6.8 Interrupt Enable Register (Interrupt_Enable)....594
6.6.6.9 Missed Frame and Buffer Overflow Counter Register (Missed_Frame_And_Buffer_Overflow_Counter) 594
6.6.6.10 Receive Interrupt Watchdog Timer Register (Receive_Interrupt_Watchdog_Timer)....594
6.6.6.11 AXI Bus Mode Register (AXI_Bus_Mode)....594
6.6.6.12 AXI Status Register (AXI_Status)....594
6.6.6.13 Current Host Transmit Descriptor Register (Current_Host_Transmit_Descriptor) .594
6.6.6.14 Current Host Receive Descriptor Register (Current_Host_Receive_Descriptor)...594
6.6.6.15 Current Host Transmit Buffer Address Register (Current_Host_Transmit_Buffer_Address) 595
6.6.6.16 Current Host Receive Buffer Address Register (Current_Host_Receive_Buffer_Address) 595
6.6.6.17 HW Feature Register (HW_Feature) 595

6.6.7 MAC Register 596
6.6.7.1 MAC Configuration Register (MAC_Configuration) 596
6.6.7.2 MAC Frame Filter Register (MAC_Frame_Filter) 596
6.6.7.3 GMII Address Register (GMII_Address) 596
6.6.7.4 GMII Data Register (GMII_Data) 597
6.6.7.5 Flow Control Register (Flow_Control) 597
6.6.7.6 VLAN Tag Register (VLAN_Tag) 597
6.6.7.7 Version Register (Version) 597
6.6.7.8 Debug Register (Debug) 597
6.6.7.9 LPI Control and Status Register (LPI_Control_Status) 597
6.6.7.10 LPI Timers Control Register (LPI_Timers_Control) 597
6.6.7.11 Interrupt Status Register (Interrupt_Status) 597
6.6.7.12 Interrupt Mask Register (Interrupt_Mask) 598
6.6.7.13 MAC Address0 High Register (MAC_Address0_High) 598
6.6.7.14 MAC Address0 Low Register (MAC_Address0_Low) 598
6.6.7.15 MAC Address1 High Register (MAC_Address1_High) 598
6.6.7.16 MAC Address1 Low Register (MAC_Address1_Low) 598
6.6.7.17 Watchdog Timeout Register (WDog_Timeout) 598
6.6.7.18 Hash Table Register 0 (Hash_Table_Reg0) 599
6.6.7.19 VLAN Hash Table Register (VLAN_Hash_Table_Reg) 599
6.6.8 IEEE 1588 Timestamp Registers 600
6.6.8.1 Timestamp Control Register (Timestamp_Control) 600
6.6.8.2 Sub Second Increment Register (Sub_Second_Increment) 600
6.6.8.3 System Time – Seconds Register, Nanoseconds Register (System_Time_Seconds, System_Time_Nanoseconds) 600
6.6.8.4 System Time – Seconds Update Register, Nanoseconds Update Register (System_Time_Seconds_Update, System_Time_Nanoseconds_Update) 601
6.6.8.5 Timestamp Addend Register (Timestamp_Addend) 601
6.6.8.6 Target Time Seconds Register, Target Time Nanoseconds Register (Target_Time_Seconds, Target_Time_Nanoseconds) 601
6.6.8.7 Timestamp Status Register (Timestamp_Status) 601
6.6.8.8 PPS Control Register (PPS_Control) 601
6.6.8.9 Auxiliary Timestamp – Nanoseconds Register (Auxiliary_Timestamp_Nanoseconds) 601
6.6.8.10 PPS0 Interval Register (PPS0_Interval) 601
6.6.8.11 PPS0 Width Register (PPS0_Width) 601
6.6.8.12 PPS1 Target Time Seconds Register (PPS1_Target_Time_Seconds) 602
6.6.8.13 PPS1 Target Time Nanoseconds Register (PPS1_Target_Time_Nanoseconds) 602

Section 7 HSR Switch 603
7.1 Overview 603
7.1.1 HSR RedBox 605
7.2 Signal Interface 606
7.3 Register Map 607
7.4 Register Description 609
7.4.1 RCI_WCONFIG — Send Config Register 609
7.4.2 RCI_RCONFIG — Receive Config Register 610
7.4.3 RCI_INT — Interrupt Register 611
7.4.4 RCI_TXRX_DATA[n] — Frame Data Send / Receive Register (n = 0..508) 612

7.4.5 R_MACADL — Own MAC Address Low Register 612
7.4.6 R_MACADH — Own MAC Address High Register....613
7.4.7 R_TST1 — Alive Test 1 Register....613
7.4.8 R_TST2 — Alive Test 2 Register....614
7.4.9 R_PEN — Port Enable Register 615
7.4.10 R_PNT_AGT — Proxynode Table Aging Time Register 617
7.4.11 R_DD_AGT — Duplicate Detection Aging Time Register 617
7.4.12 R_MACFLT_I[m]L — Filter MAC Address Interlink [m] Low Register (m = 1 to 8)....618
7.4.13 R_MACFLT_I[m]H — Filter MAC Address Interlink [m] High Register (m = 1 to 8) 618
7.4.14 R_MACFLT_C[m]L — Filter MAC Address CPU [m] Low Register (m = 1 to 6)....619
7.4.15 R_MACFLT_C[m]H — Filter MAC Address CPU [m] High Register (m = 1 to 6)....619
7.4.16 R_VER — HSR Version Register 620
7.4.17 R_RAM_STA — RAM Status Register 620
7.4.18 R_UFMC — Used Frame Memory Count Register 621
7.4.19 R_FRA_ALL_ARX — Received Frame A Register 621
7.4.20 R_FRA_TAG_ARX — Received Tagged Frame A Register 622
7.4.21 R_FRA_NLL_ARX — Received Not linklocal Frame A Register....622
7.4.22 R_FRA_ERR_ARX — Received Error Frame A Register 623
7.4.23 R_FRA_WRO_ARX — Wrong LAN Frame A Register 623
7.4.24 R_FRA_ALL_ATX — Sent Frame A Register 624
7.4.25 R_FRA_TAG_ATX — Sent Tagged Frame A Register 624
7.4.26 R_FRA_NLL_ATX — Sent Not Linklocal Frame A Register 625
7.4.27 R_FRA_ALL_BRX — Received Frame B Register 625
7.4.28 R_FRA_TAG_BRX — Received Tagged Frame B Register 626
7.4.29 R_FRA_NLL_BRX — Received Not Linklocal Frame B Register 626
7.4.30 R_FRA_ERR_BRX — Received Error B Register....627
7.4.31 R_FRA_WRO_BRX — Wrong LAN Count B Register 627
7.4.32 R_FRA_ALL_BTX — Sent Frame B Register 628
7.4.33 R_FRA_TAG_BTX — Sent Tagged Frame B Register 628
7.4.34 R_FRA_NLL_BTX — Sent Not Linklocal Frame B Register 629
7.4.35 R_FRA_ALL_CRX — Received Frame C Register 629
7.4.36 R_FRA_TAG_CRX — Received Tagged Frame C Register....630
7.4.37 R_FRA_NLL_CRX — Received Not Linklocal Frame C Register 630
7.4.38 R_FRA_ERR_CRX — Received Error C Register 631
7.4.39 R_FRA_WRO_CRX — Wrong LAN Count C Register....631
7.4.40 R_FRA_ALL_CTX — Sent Frame C Register 632
7.4.41 R_FRA_TAG_CTX — Sent Tagged Frame C Register....632
7.4.42 R_FRA_NLL_CTX — Sent Not Linklocal Frame C Reg 633
7.4.43 R_FREE_FRA_M — Free Frame Memory Count 633
7.4.44 R_DBG_RPT1 — Internal Debug Reports Register 1 634
7.4.45 R_DBG_RPT2 — Internal Debug Reports Register 2 635
7.4.46 R_PNT_S — Proxynodetable Status Register 637
7.4.47 R_PNT_D — Proxynodetable Data Register....638

7.4.48 RPTP_ID — PTP Core ID Register 638
7.4.49 RPTP_TST — PTP Core Test Register 639
7.4.50 RPTP_VER — PTP Version Register 639
7.4.51 RPTP_GPO — General Purpose Register 640
7.4.52 RPTP_INT — PTP Interrupt Register 641
7.4.53 RPTP_INT_MSK — PTP Interrupt Mask Register 642
7.4.54 RPTP_TS_STAT_[m] — Timestamp Status Register P[m] (m = 1 to 4) 643
7.4.55 RPTP_TS_RD_[m] — Timestamp Read Register P[m] (m = 1 to 4) 644
7.4.56 RPTP_PORT_CONF_[m] — Port Config Register P[m] (m = 1 to 4) 645
7.4.57 RPTP_P_DELAY_[m] — Peer Delay Register P[m] (m = 1 to 4) 645
7.4.58 RPTP_PHY_DLY_TX_[m] — PHY Tx Delay Register P[m] (m = 1 to 4) 646
7.4.59 RPTP_PHY_DLY_RX_[m] — PHY Rx Delay Register P[m] (m = 1 to 4) 646
7.4.60 RPTP_BUF_STAT — Buffer Status Register 647
7.5 Operation 648
7.5.1 HSR Initializing Flow 648
7.5.1.1 Initializing 648
7.5.1.2 HSR_IP_INIT 650
7.6 Usage Notes 651
7.6.1 PTP Registers Assignment 651
7.6.2 Restriction 651

Section 8 Sercos III Slave Controller 652
8.1 Overview 652
8.2 Signal Interface 654
8.3 Register Map 655
8.4 Register Description 657
8.4.1 IDR — Identification Register 657
8.4.2 GCSFR — Global Control / Status / Feature Register 658
8.4.3 IER0 — Interrupt Enable Register 659
8.4.4 IMR0 — Interrupt Multiplex Register 659
8.4.5 IRR0 — Interrupt Reset/Status Register 660
8.4.6 DFCSR — Data Flow Control/Status Register 661
8.4.7 PHASESR — Phase Status Register 662
8.4.8 TGSR1 — Telegram Reset/Status Register Port 1 663
8.4.9 TGSR2 — Telegram Reset/Status Register Port 2 664
8.4.10 DESCR — Descriptor Control Register 665
8.4.11 STRBR — System Timer Read Back Register 665
8.4.12 TCSR — Timing Control/Status Register 666
8.4.13 TRDLY — Ring Delay Register 667
8.4.14 TDMST1 — Time Delay MST Port 1 667
8.4.15 TDMST2 — Time Delay MST Port 2 668
8.4.16 SCR1 — Sync Time Register Port 1 668
8.4.17 SCR2 — Sync Time Register Port 2 669

8.4.18 SVCCSR — SVC Control / Status 669
8.4.19 DTDIVCLK —Delay Time for DIV_CLK 670
8.4.20 TDIV_NDIVCLK — DIV_CLK Time / Count Register 670
8.4.21 S3LED — Sercos III LED Control 671
8.4.22 WDCSR — Watchdog Control & Status 672
8.4.23 WDCNT — Watchdog Counter 672
8.4.24 SFCR — Sercos Frame Control 673
8.4.25 MIICSR — MDIO Control / Status Register 674
8.4.26 DBGOCR — Debug Output Control 675
8.4.27 SEQCNT — Sequence Counter 675
8.4.28 MAC1P1_0 — MAC Address 0 676
8.4.29 MAC1P1_1 — MAC Address 1 676
8.4.30 IPCSR1 — IP Status/Control register port 1 677
8.4.31 IPCSR2 — IP Status/Control register port 2 678
8.4.32 IPRRS1 — IP Rx Ram Segment Port 1 679
8.4.33 IPRRS2 — IP Rx Ram Segment Port 2 679
8.4.34 IPRXS1 — IP Receive Stack Port 1 680
8.4.35 IPRXS2 — IP Receive Stack Port 2 680
8.4.36 IPTXS1 — IP Transmit Stack Port 1 681
8.4.37 IPTXS2 — IP Transmit Stack Port 2 681
8.4.38 IPLASTFL — Remaining Frame Length 682
8.4.39 IPFRXOK — IP aFramesReceivedOK 682
8.4.40 IPFTXOK — IP aFramesTransmittedOK 683
8.4.41 IPFCSERR — IP aFCSErrors 683
8.4.42 IPALGNERR — IP aAlignmentErrors 684
8.4.43 IPDISRXB — IP aDiscardResRxBuf 684
8.4.44 IPDISCLB — IP aDiscardResColBuf 685
8.4.45 IPCHVIOL — IP aIPChannelViolation 685
8.4.46 aSercosErrorCount — Sercos III Error Counter 686
8.4.47 MSTLMAX — Maximum Sequential MST Losses 686
8.4.48 MSTLSUM — Sum MST Losses 687
8.4.49 MSTLACT — Actual Sequential MST Losses 687
8.4.50 TMDSCL — Timing Descriptor Lower 688
8.4.51 TMDSCU — Timing Descriptor Upper 688
8.4.52 TMDSCSEL — Timing Descriptor Select 689
8.4.53 PTMDSCL — Port Timing Descriptor Lower 689
8.4.54 PTMDSCU — Port Timing Descriptor Upper 690
8.4.55 PTMDSCSEL — Port Timing Descriptor Select 690
8.4.56 STNS — System Time (nanoseconds) 691
8.4.57 STSEC — System Time (seconds) 691
8.4.58 STNSTSR — System Time TSRef (nanoseconds) 692
8.4.59 STSECTSR — System Time TSRef (seconds) 692
8.4.60 SCCAB — Subcycle Counter Control & Status 693

8.4.61 SCCMDT — Subcycle Counter MDT 693
8.4.62 RXBUF[n]_P1A — Receive Buffer [n] Base Address for Port 1 and Buffer System A (n = 0..2) 694
8.4.63 RXBUF[n]_P1B — Receive Buffer [n] Base Address for Port 1 and Buffer System B (n = 0..2) 694
8.4.64 RXBUF_P1SVC — Receive Buffer Base Address for Service Channel Data at Port 1 695
8.4.65 RXBUF[n]_P2A — Receive Buffer [n] Base Address for Port 2 and Buffer System A (n = 0..2) 695
8.4.66 RXBUF[n]_P2B — Receive Buffer [n] Base Address for Port 2 and Buffer System B (n = 0..2) 696
8.4.67 RXBUF_P2SVC — Receive Buffer Base Address for Service Channel Data at Port 2 696
8.4.68 TXBUF[n]_A — Transmit Buffer [n] Base Address for Buffer System A (n = 0..3) 697
8.4.69 TXBUF[n]_B — Transmit Buffer [n] Base Address for Buffer System B (n = 0..3) 697
8.4.70 TXBUF_P1 — Transmit Buffer Base Address for Port 1 Only 698
8.4.71 TXBUF_P2 — Transmit Buffer Base Address for Port 2 Only 698
8.4.72 TXBUF_SVC — Transmit Buffer Base Address for Service Channel Data 699
8.4.73 RXBUFCSR_A — Receive Buffer Control Buffer System A 700
8.4.74 RXBUFTV_A — Rx Buffer Telegram Valid A 701
8.4.75 RXBUFTR_A — Rx Buffer Telegram Requirements A 702
8.4.76 TXBUFCSR_A — Transmit Buffer Control for Buffer System A 703
8.4.77 RXBUFCSR_B — Receive Buffer Control Buffer System B 704
8.4.78 RXBUFTV_B — Rx Buffer Telegram Valid B 705
8.4.79 RXBUFTR_B — Rx Buffer Telegram Requirements B 706
8.4.80 TXBUFCSR_B — Transmit Buffer Control for Buffer System B 707
8.5 Operation 708
8.5.1 Initializing Flow 708
8.5.1.1 Initializing 708
8.5.1.2 SERCOS3_IP_INIT 709
8.5.1.3 SERCOS3_DSC_INIT 710
8.5.1.4 SERCOS3_TCNT_INIT_START 711
8.5.2 Communication Flow 712
8.5.2.1 Transmit 712
8.5.2.2 Receive 712
8.5.3 Interrupt Control 713
8.5.4 Slave Timing Control 715
8.5.4.1 Timing Control and Status Register (TCSR) 716
8.5.4.2 System Timer Read Back Register (STRBR) 716
8.5.4.3 Central Timer TCNT 717
8.5.4.4 Communication Timers TCNT[Port 1 / Port 2] 718
8.5.4.5 System Timer 721
8.5.4.6 Subcycle Counters 722
8.5.5 Slave MST Processing 723
8.5.5.1 Register Interface for Slave MST Processing 724
8.5.6 DIV_CLK Function 725
8.5.6.1 DIV_CLK mode 0 — n Times Within One Communication Cycle 725
8.5.6.2 DIV_CLK Mode 1 — Once after N Communication Cycle 726

8.5.7 Data Flow Control 727
8.5.7.1 Data Flow Control and Status Register (DFCSR) 727
8.5.7.2 Descriptor Control Register (DESCR) 727
8.5.7.3 Sequence Counter Register (SEQCNT) 728
8.5.7.4 Telegram Status Register (TGSR1/TGSR2) 728
8.5.7.5 Supervising the Communication Phase — Phase Status Register (PHASESR) ...729
8.5.8 Tx MAC Operation 730
8.5.8.1 Principle Structure of Tx MAC 730
8.5.8.2 Sources of Telegram Data in AT 730
8.5.8.3 Tx Descriptors 731
8.5.8.4 Tx RAM Pointer 732
8.5.8.5 Overview of Tx Descriptor Engine 732
8.5.8.6 Tx MAC Processing Description 733
8.5.8.7 Tx MAC Quadbuffering 733
8.5.9 Rx MAC Operation 734
8.5.9.1 Rx Descriptors 734
8.5.9.2 Rx RAM Pointer 735
8.5.9.3 Rx MAC Triple Buffering 735
8.5.10 Slave Service Channel Operation 736
8.5.10.1 SVC Control & Status Register (SVCCSR) 736
8.5.10.2 Service Container 737
8.5.10.3 Service Container Pointers 737
8.5.10.4 Service Container Control Words 739
8.5.10.5 Service Processor Operation Details 740
8.5.10.6 SVC Telegram Data 742
8.5.10.7 Byte Order of Data in the Telegram 743
8.5.11 Hardware Watchdog 743
8.5.12 MST Monitoring 744
8.5.13 IP Channel Operation 745
8.5.13.1 MAC Address Register 745
8.5.13.2 IP Control and Status Register (IPCSR 1/2) 745
8.5.13.3 Collision Buffer Operation 746
8.5.13.4 Transmitting IP Telegrams 747
8.5.13.5 Receiving IP Frames 749
8.5.13.6 Remaining Frame Length (IPLASTFL) 751
8.5.13.7 Frame and Error Counters 752
8.5.14 MDIO Control 753
8.5.15 Debug Output Control 753
8.5.16 Sercos III LED Control 754
8.5.16.1 S3LED 754
8.5.16.2 Active LED 754

Section 9 R-IN Engine Accessory Register 755
9.1 Register Map 755
9.2 Register Description 756
9.2.1 IDCODE — IDCODE Register 756
9.2.2 SCRATCH[n] — Scratch register [n] (n = 0..3) 756
9.2.3 RINSPCMD — R-IN System Protect Command Register 757
9.2.4 RTOSRST — HW-RTOS and HW-RTOS GMAC Reset Register 757

9.3 Usage Notes 758
9.3.1 R-IN Engine Accessory Register 758
9.3.1.1 R-IN System Protect Command Register (RINSPCMD) 758
9.3.1.2 HW-RTOS Reset Register (RTOSRST) 759

Section 10 Ethernet Accessory Register 760
10.1 Register Map 760
10.2 Register Description 761
10.2.1 PRCMD — Ethernet Protect Register 761
10.2.2 ESIDCODE — Ethernet Peripherals IDCODE Register 761
10.2.3 MODCTRL — Mode Control Register 762
10.2.4 PTPMCTRL — PTP Mode Control Register 762
10.2.5 PHYLNK — Ethernet PHY Link Mode Register 763
10.2.6 PTCTRL — PortTrigger Control Register 764
10.2.7 DMACTRL — DMAC Control Register 765
10.2.8 CONVCTRL[m] — RGMII/RMII Converter[m] Control Register (m = 1 to 5) 766
10.2.9 CONVRST — RGMII/RMII Converter Reset Control Register 767
10.2.10 ECATOFFADR — EtherCAT PHY Offset Address Register 768
10.2.11 ECATOPMOD — EtherCAT Operation Mode Register 768
10.2.12 ECATDBGC — EtherCAT Debug Control Register 769
10.2.13 SCINTCON — SERCOS Interrupt Control Register 770
10.2.14 SWCTRL — A5PSW Control Register 771
10.2.15 SWDUPC — A5PSW Duplex Mode Register 772
10.2.16 RMTAGCTRL — HW-RTOS GMAC Management TAG Control Register 772
10.2.17 HSRMOD — HSR Mode Indication Register 773
10.3 Operation 774
10.4 Usage Notes 775
10.4.1 Ethernet Protect Register 775
10.4.2 PortTrigger Control Register 777
10.4.3 Ethernet PHY Link Mode Register 777
10.4.4 DMAC Control Register 778

Section 11 MDIO Interfaces 779
11.1 Overview 779
11.2 Signal Interface 781
11.3 Operation 782

Section 1 R-IN Engine Outline

1.1 Overview

Table 1.1 R-IN Engine(Core, RAM) for RZ/N1

Item	Description
CPU core	Arm® 32-bit CPU Cortex® -M3 (Revision r2p1) • Maximum operating frequency: 125 MHz
Nested Vectored Interrupt Controller (NVIC)	• 128 interrupts
On-chip debug function	CoreSight® Interface
Internal SRAM	Instruction 1 MByte (RAM w/ECC), Data 1 MByte (RAM w/ECC)
Internal System Bus	• 32-bit system bus at maximum 125 MHz • 128-bit communication bus at maximum 125 MHz
Hardware Real-Time OS accelerator (HW-RTOS)	• RTOS system call • Task scheduling and tick offloading • Hardware function manager for Buffer allocator, Internal DMA controller, HW-RTOS GMAC • Hardware Interrupt service routine (HWISR)
Dedicated Gigabit-Ether MAC (HW-RTOS GMAC)	• 1 channel • GMII interface • Ethernet accelerator function • 128 KByte Buffer RAM (shared with HSR)

Table 1.2 Ethernet Peripherals for RZ/N1

Item	Description
Independent Gigabit-Ether MAC (GMAC)	• 2 channels • GMII / MII interface
HSR (Optional)*1	HSR compliant to IEC62439-3 section5 • 3 ports (includes an interlink port) • 144 KByte Frame buffer (shared with HW-RTOS GMAC)
Advanced 5 Port Switch (A5PSW)	• 5 ports (includes a management port) • PRP functionality (Optional)*1 • Hub function
SercosIII Slave Controller	• 2 ports
EtherCAT Slave Controller	• 3 ports
RGMII/RMII Converter	• MII of MAC from/to RMII of PHY • MII of MAC from/to RGMII of PHY • GMII of MAC from/to RGMII of PHY • MII of MAC from/to MII of PHY (through mode)

Note 1. For the supported products, refer to *Section 1.4, List of Products*, in the *RZ/N1D Group, RZ/N1S Group, RZ/N1L Group User's Manual: System Introduction, Multiplexing, Electrical and Mechanical Information*.

Table 1.3 Normative Reference

Standard Name	Number of Standard
Ethernet	IEEE802.3
EtherCAT®	IEC 62407
Sercos®III	IEC 61491
HSR	IEC 62439-3 Clause 5
PRP	IEC 62439-3 Clause 4
PTP	IEEE1588

Figure 1.1 R-IN Engine and Ethernet Peripherals Block Diagram

Section 2 Hardware Real Time OS (HW-RTOS)

2.1 Overview

The Hardware Real time OS supports 30 types of system calls including event, semaphore and mailbox.

- ITRON like system calls
  - 30 system calls for elements such as events, semaphores, and mailboxes
- Task Scheduler
  - Hardware ISR: Maximum 32 selectable from 128 interrupts
  - Context: 64
  - Semaphore: 128
  - Event: 64
  - Mailboxes: 64
  - Mailbox elements: 192
  - Context priorities: 16
- Hardware function manager

CAUTION

HW-RTOS can't be accessed from Cortex-A7.

Section 3 Gigabit Ethernet MAC (HW-RTOS GMAC)

3.1 Overview

The R-IN Engine incorporates a Gigabit Ethernet MAC (HW-RTOS GMAC).

- Number of ports: 1
- 1000 Mbps
- Full-duplex communication
- Management tag control function for Advanced 5 port Switch (A5PSW)

CAUTION

HW-RTOS GMAC must be set 1 Gbps and full-duplex mode because GMII is used for connection to A5PSW.
HW-RTOS GMAC can't be accessed from Cortex-A7.

Figure 3.1 HW-RTOS GMAC Block Diagram

3.2 Signal Interface

Signal Name	I/O	Description	Active
Clock			
HWRTOS_CLK	I	HW-RTOS clock	
HWRTOS_MDCCLK	I	HW-RTOS GMAC MDC clock	
Related Clock			
RINBUS_HCLK	—	R-IN Engine clock. Since HW-RTOS GMAC belongs to R-IN Engine, this clock should be supplied.	
Interrupt			
HWRTOS_ETHTFIU_Int	O	HW-RTOS GMAC TX FIFO underflow, pulse sensitive	High
HWRTOS_ETHTFIE_Int	O	HW-RTOS GMAC TX FIFO error interrupt, pulse sensitive	High
HWRTOS_ETHDTIE_Int	O	HW-RTOS GMAC MACDMA transmission error, pulse sensitive	High
HWRTOS_ETHDMAIT_Int	O	HW-RTOS GMAC MACDMA transmission completion, pulse sensitive	High
HWRTOS_ETHIT_Int	O	HW-RTOS GMAC transmission completion interrupt, pulse sensitive	High
HWRTOS_ETHDMAIR_Int	O	HW-RTOS GMAC MACDMA reception completion, level sensitive	High
HWRTOS_ETHDRIE_Int	O	HW-RTOS GMAC MACDMA reception error, pulse sensitive	High
HWRTOS_ETHRFI_Int	O	HW-RTOS GMAC MACDMA valid frame reception completion, level sensitive	High
HWRTOS_ETHRFE_Int	O	HW-RTOS GMAC MACDMA error frame reception completion, level sensitive	High
HWRTOS_ETHRFIV_Int	O	HW-RTOS GMAC RX FIFO overflow, pulse sensitive	High
HWRTOS_ETHMMAI_Int	O	HW-RTOS GMAC MII management access completion interrupt, pulse sensitive	High
HWRTOS_ETHPPIT_Int	O	HW-RTOS GMAC pause packet transmission completion, pulse sensitive	High
HWRTOS_BUFDMA_Int	O	HW-RTOS GMAC InterBuffer DMA transfer completion, pulse sensitive	High
HWRTOS_BUFDMAERR_Int	O	HW-RTOS GMAC InterBuffer DMA error, pulse sensitive	High
HWRTOS_BRAMERR_Int	O	HW-RTOS GMAC Buffer RAM area access error, pulse sensitive	High

3.3 Register Map

3.3.1 HW-RTOS HWFC (Hardware Function Call) Register Map

Table 3.1 HW-RTOS HWFC (Hardware Function Call) Register Map

Address	Register Symbol	Register Name
400E 0000h	C0TYPE	Hardware function type register
400E 0008h	C0STAT	Hardware function status register
400E F000h	SYSC	Hardware function system call register
400E F004h	R4	Hardware function argument register 4
400E F008h	R5	Hardware function argument register 5
400E F00Ch	R6	Hardware function argument register 6
400E F010h	R7	Hardware function argument register 7
400E F014h	CMD	Hardware function command register
400E F020h	R0	Hardware function return value register 0
400E F024h	R1	Hardware function return value register 1

3.3.2 HW-RTOS GMAC (Gigabit Ethernet MAC) Register Map

Table 3.2 HW-RTOS GMAC (Gigabit Ethernet MAC) Register Map

Address	Register Symbol	Register Name
400F 000Ch	GMAC_TXID	TX ID register
400F 0010h	GMAC_TXRESULT	TX result register
400F 0020h	GMAC_MODE	Mode register
400F 0024h	GMAC_RXMODE	RX mode register
400F 0028h	GMAC_TXMODE	TX mode register
400F 0030h	GMAC_RESET	Reset register
400F 0080h + 4h × (m − 1)	GMAC_PAUSE[m] (m = 1 to 5)	Pause packet data register [m] (not available)
400F 0098h	GMAC_FLWCTL	RX flow control register (not available)
400F 009Ch	GMAC_PAUSPKT	Pause packet register (not available)
400F 00A0h	GMAC_MIIM	MIIM register
400F 0100h + 8h × (m − 1)	GMAC_ADR[m]A (m = 1 to 16)	MAC address register [m]A
400F 0104h + 8h × (m − 1)	GMAC_ADR[m]B (m = 1 to 16)	MAC address register [m]B
400F 0200h	GMAC_RXFIFO	RX FIFO status register
400F 0204h	GMAC_TXFIFO	TX FIFO status register
400F 0208h	GMAC_ACC	TCP/IPACC register
400F 0220h	GMAC_RXMAC_ENA	RX MAC enable register
400F 1100h	BUFID	Receive Buffer information register

3.4 Register Description

3.4.1 HW-RTOS HWFC (Hardware Function Call) Register Description

3.4.1.1 C0TYPE — Hardware Function Type Register

Address: 400E 0000h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	C0TYPEB															
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	C0TYPEB															
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Table 3.3 C0TYPE Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	C0TYPEB	These bits set the type of hardware function.	R/W

3.4.1.2 C0STAT — Hardware Function Status Register

Address: 400E 0008h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	C0STATB															
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	C0STATB															
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Table 3.4 C0STAT Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	C0STATB	These bits set the status of hardware function.	R/W

3.4.1.3 SYSC — Hardware Function System Call Register

When a command is written to the SYSC register, the corresponding function is executed.

Address: 400E F000h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SYSC															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 3.5 SYSC Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b16	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b0	SYSC	Specifies the hardware function to be used. One of the following functions can be specified: 5000h: Acquires a long buffer. 5006h: Acquires a short buffer. 5001h: Release the whole area of the buffer. 5002h: Release the part of the buffer. 5101h: Enables DMA for the reception MAC. 5102h: Disables DMA for the reception MAC. 510Bh: Controls reception MACDMAC interrupts. 510Dh: Obtains the error source in reception MACDMAC. 5100h: Starts transfer of reception MACDMAC. 510Ch: Obtains the error source in transmission MACDMAC. 5211h: Starts DMA transfer between the buffer RAM and data RAM. 5212h: Starts replacing data in the buffer RAM or data RAM. 5104h: Starts DMA transfer between the buffer RAM and buffer RAM. 5114h: Starts DMA transfer between the buffer RAM and buffer RAM. (Descriptor) Other: Setting prohibited	R/W

3.4.1.4 R4 — Hardware Function Argument Register 4

Address: 400E F004h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	R4B															
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	R4B															
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Table 3.6 R4 Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	R4B	Specifies the argument to be passed to the hardware function.	R/W

3.4.1.5 R5 — Hardware Function Argument Register 5

Address: 400E F008h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	R5B															
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	R5B															
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Table 3.7 R5 Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	R5B	Specifies the argument to be passed to the hardware function.	R/W

3.4.1.6 R6 — Hardware Function Argument Register 6

Address: 400E F00Ch

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	R6B															
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	R6B															
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Table 3.8 R6 Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	R6B	Specifies the argument to be passed to the hardware function.	R/W

3.4.1.7 R7 — Hardware Function Argument Register 7

Address: 400E F010h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	R7B															
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	R7B															
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Table 3.9 R7 Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	R7B	Specifies the argument to be passed to the hardware function.	R/W

3.4.1.8 CMD — Hardware Function Command Register

This register is used to enable hardware functions.

Address: 400E F014h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	CMDB															
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CMDB															
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Table 3.10 CMD Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	CMDB	These bits set the commands for hardware functions. 0x0000 8004: Enable hardware function and hardware real-time OS.	R/W

3.4.1.9 R0 — Hardware Function Return Value Register 0

Address: 400E F020h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	R0B															
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	R0B															
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Table 3.11 R0 Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	R0B	Return value of the hardware function call	R/W

3.4.1.10 R1 — Hardware Function Return Value Register 1

Address: 400E F024h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	R1B															
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	R1B															
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Table 3.12 R1 Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	R1B	Return value of the hardware function call	R/W

3.4.2 HW-RTOS GMAC (Gigabit Ethernet MAC) Register Description

3.4.2.1 GMAC_TXID — TX ID Register

Address: 400F 000Ch

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TXID															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TXID															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 3.13 GMAC_TXID Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	TXID	Indicates the ID of the transmission frame for the GMAC_TXRESULT register.	R

3.4.2.2 GMAC_TXRESULT — TX Result Register

Address: 400F 0010h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	TCMP	TABT	—	—	—	—	OVERFW	UNDERFW	—	—	—	—	—	FIFOUFLOW
Value after reset	X	X	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 3.14 GMAC_TXRESULT Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b14	Reserved	These bits are read as 0.	R
b13	TCMP	Transmission Finished	R
b12	TABT	Transmission Abort Occurred.	R
b11 to b8	Reserved	These bits are read as 0.	R
b7	OVERFW	A frame longer than 1,518 octets was written to the transmit FIFO.	R
b6	UNDERFW	A frame shorter than the minimum frame length was written to the transmit FIFO.	R
b5 to b1	Reserved	These bits are read as 0.	R
b0	FIFOUFLOW	An FIFO underflow occurred during transmission.	R

3.4.2.3 GMAC_MODE — Mode Register

The GMAC_MODE register is used to control the operating mode of HW-RTOS GMAC.

Address: 400F 0020h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	ETHMODE	DUPMODE	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Table 3.15 GMAC_MODE Register Contents

Bit Position	Bit Name	Function	R/W
b31	ETHMODE	1: HW-RTOS GMAC operates in Giga bit Ethernet mode. Use this mode in order to connect to Advanced 5 port Switch. 0: Don’t use	R/W
b30	DUPMODE	1: HW-RTOS GMAC operates in Full Duplex mode. Use this mode in order to connect to Advanced 5 port Switch. 0: Don’t use	R/W
b29 to b0	Reserved	These bits are read as 0. The write value should be 0.	R

3.4.2.4 GMAC_RXMODE — RX Mode Register

The GMAC_RXMODE register is used to control frame reception. The RX FIFO treats a word as 64-bit, and the FIFO size is 4 KB.

Address: 400F 0024h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	AFILLTEREN	MFILLTEREN	SFRXFIFO	RAMASKEN	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	1	0	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	REMPTH		RFULLTH		RRTTH			—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	X	X	X	X	X	X	X	X	X

Table 3.16 GMAC_RXMODE Register Contents (1/2)

Bit Position	Bit Name	Function	R/W
b31	AFILLTEREN	1: Enables address Filtering*1 0: Receives all frames by means of ignoring destination address in a frame.	R/W
b30	MFILLTEREN	1: Discards multicast address frames that are not registered in MAC address registers (GMAC_ADR[m]A and GMAC_ADR[m]B). (m = 1 to 16) 0: Receives all multicast address frames	R/W
b29	SFRXFIFO	1: Store & Forward mode The receive DMA controller does not start operation until data up to the end of the frame is written to RX FIFO. 0: Cut Through mode The receive DMA controller starts operation when the number of data words in RX FIFO reaches the number of words set by RRTTH bit.	R/W
b28	RAMASKEN	1: Enables the function that can be set by the BITMSK[7:0] bits of the GMAC_ADR[m]B register (comparative mask function for Destination Address[7:0]). (m = 1 to 16) 0: Disables the above function	R/W
b27 to b16	Reserved	These bits are read as 0. The write value should be 0.	R
b15, b14	REMPTH	When the number of data words in the FIFO is below this value, the REMP bit of the GMAC_RXFIFO register is set to “1”. 00b: 4 words 01b: 8 words 10b: 16 words 11b: 32 words	R/W
b13, b12	RFULLTH	When the empty space in the FIFO is below this value, the RFULL bit in the GMAC_RXFIFO register becomes “1”. 00b: 4 words 01b: 8 words 10b: 16 words 11b: 32 words	R/W

Table 3.16 GMAC_RXMODE Register Contents (2/2)

Bit Position	Bit Name	Function	R/W
b11 to b9	RRTTH	If the number of data words in the FIFO exceeds this value, the RRT bit of the GMAC_RXFIFO register is set to “1”. 000b: 4 words 001b: 8 words 010b: 16 words 011b: 32 words 100b: 64 words 101b: 128 words 110b: 256 words 111b: 512 words	R/W
b8 to b0	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. Even though address filtering is enabled and MAC address is not matched, MAC Control Frame (e.g. Pause Packet) is always received. MAC Control Frame is the frame with the destination address 01-80-C2-00-00-01.

3.4.2.5 GMAC_TXMODE — TX Mode Register

The GMAC_TXMODE register is used to control frame transmission. The TX FIFO treats a word as 64-bit, and the FIFO size is 4 KB.

Address: 400F 0028h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	LPTXEN	SF	SPTXEN	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	TEMPTH			TFULLTH		—	TRBMODE		—	—	—	—	—	—
Value after reset	X	X	0	0	0	0	0	X	0	0	X	X	X	X	X	X

Table 3.17 GMAC_TXMODE Register Contents

Bit Position	Bit Name	Function	R/W
b31	Reserved	These bits are read as 0. The write value should be 0.	R/W
b30	LPTXEN	1: Permits transmission of frames that are longer than the length prescribed by IEEE802.3. Set this bit if a frame includes VLAN Tag or Management Tag. 0: Prohibits transmission of frames that are longer than the length prescribed by IEEE802.3.	R/W
b29	SF	1: Starts transmission after the data up to the end of the frame is written to TX FIFO. Should be set to 1. 0: Don't use.	R/W
b28	SPTXEN	1: Permits transmission of frames that are shorter than the length prescribed by IEEE802.3. 0: Prohibits transmission of frames that are shorter than the length prescribed by IEEE802.3.	R/W
b27 to b14	Reserved	These bits are read as 0. The write value should be 0.	R
b13 to b11	TEMPTH	If fewer words of data are in the TX FIFO buffer than the value specified by these bits, the TEMP bit in the GMAC_TXFIFO register becomes 1. 000b: 4 words 001b: 8 words 010b: 16 words 011b: 32 words 100b: 64 words 101b: 128 words 110b: 256 words 111b: 512 words	R/W
b10, b9	TFULLTH	If the empty space in the TX FIFO buffer is below the value specified by these bits, the TFULL bit in the GMAC_TXFIFO becomes 1. 00b: 4 words 01b: 8 words 10b: 16 words 11b: 32 words	R/W
b8	Reserved	These bits are read as 0. The write value should be 0.	R
b7, b6	TRBMODE	Controls how to write the transmission result to the GMAC_TXRESULT register. 00b: The result is always written. 01b: The result is written only when an error occurred. 10b: The result is never written. Others: Setting prohibited	R/W
b5 to b0	Reserved	These bits are read as 0. The write value should be 0.	R

3.4.2.6 GMAC_RESET — Reset Register

Address: 400F 0030h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	ALLRST	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TXRST	RXRST	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Table 3.18 GMAC_RESET Register Contents

Bit Position	Bit Name	Function	R/W
b31	ALLRST	Resets HW-RTOS GMAC module which includes HW-RTOS GMAC registers. 0: Performs nothing. 1: Resets the modules.	W
b30 to b16	Reserved	These bits are read as 0. The write value should be 0.	R
b15	TXRST	Resets TX MAC and TX FIFO modules. 0: Performs nothing. 1: Resets the modules.	W
b14	RXRST	Resets RX MAC and RX FIFO modules. 0: Performs nothing. 1: Resets the modules.	W
b13 to b0	Reserved	These bits are read as 0. The write value should be 0.	R

3.4.2.7 GMAC_PAUSE[m] —Pause Packet Data Register [m] (m = 1 to 5)

The register is used to specify the pause packet to be sent.

Address: 400F 0080h + 4h × (m − 1)

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	PPDATA[m]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PPDATA[m]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 3.19 GMAC_PAUSE[m] Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	PPDATA[m]	Specifies 4 × m th to 4 × m − 3th bytes of the pause packet to be sent. Data is sent in LSB first order.	R/W

3.4.2.8 GMAC_FLWCTL — RX Flow Control Register

The GMAC_FLWCTL register is used to control the pause packet reception function.

Address: 400F 0098h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	PPRXEN	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Table 3.20 GMAC_FLWCTL Register Contents

Bit Position	Bit Name	Function	R/W
b31	PPRXEN	1: Enable auto broadcast suspension in response to reception of a pause packet. 0: Disable auto broadcast suspension in response to reception of a pause packet.	R/W
b30 to b0	Reserved	These bits are read as 0. The write value should be 0.	R

3.4.2.9 GMAC_PAUSPKT — Pause Packet Register

The GMAC_PAUSPKT register is used to control pause packet transmission.

When 1 is written to the PPR bit, transmission of a pause packet specified by GMAC_PAUSE[m] registers starts. The bit is automatically set to 0 following the completion of the transmission.

Address: 400F 009Ch

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	PPR	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Table 3.21 GMAC_PAUSPKT Register Contents

Bit Position	Bit Name	Function	R/W
b31	PPR	Controls Pause Packet Transmission 0: Nothing is to be done. 1: Starts pause packet transmission.	R/W
b30 to b0	Reserved	These bits are read as 0. The write value should be 0.	R

3.4.2.10 GMAC_MIIM — MIIM Register

Address: 400F 00A0h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	RWDV	PHYADDR					REGADDR				
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DATA															
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Table 3.22 GMAC_MIIM Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b27	Reserved	These bits are read as 0. The write value should be 0.	R
b26	RWDV	Read/write operation starts by writing the following value to this bit. Set also other bits at the same time. 1: Starts a write operation. 0: Starts a read operation. After a read/write operation has started, the status of the operation can be checked by reading the value of this bit.*1 1: Operation is completed (DATA bit is valid). 0: Waiting for the operation to complete.	R/W
b25 to b21	PHYADDR	Specifies the address of the PHY to be accessed. Because this is a write only bit, the value read from the bit is undefined.	W
b20 to b16	REGADDR	Specifies the register address of the PHY to be accessed. Because this is a write only bit, the value read from the bit is undefined.	W
b15 to b0	DATA	Indicates write data or read data.	R/W

Note 1. After reset is released, the RWDV bit is set to 1. However, the values held by the DATA[15:0] bits at this time are not valid. To use the RWDV bit to check the status correctly, make sure that the value is read from the bit after the operation has started.

3.4.2.11 GMAC_ADR[m]A — MAC Address Register [m]A (m = 1 to 16)

The GMAC_ADR[m]A and GMAC_ADR[m]B registers are used to set MAC addresses.

A maximum of 16 addresses can be registered. Multiple addresses can be filtered by using the BITMSK[7:0] bits of the GMAC_ADR[m]B register.

Address: 400F 0100h + 8h × (m − 1)

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	MADDR4B								MADDR3B							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MADDR2B								MADDR1B							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 3.23 GMAC_ADR[m]A Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b24	MADDR4B	Indicates the 4th byte from the beginning of the MAC address to be included.	R/W
b23 to b16	MADDR3B	Indicates the 3rd byte from the beginning of the MAC address to be included.	R/W
b15 to b8	MADDR2B	Indicates the 2nd byte from the beginning of the MAC address to be included.	R/W
b7 to b0	MADDR1B	Indicates the first byte of the MAC address to be included.	R/W

3.4.2.12 GMAC_ADR[m]B — MAC Address Register [m]B (m = 1 to 16)

Address: 400F 0104h + 8h × (m − 1)

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	BITMSK							
Value after reset	X	X	X	X	X	X	X	X	1	1	1	1	1	1	1	1

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MADDR6B								MADDR5B							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 3.24 GMAC_ADR[m]B Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b24	Reserved	When read, the value returned is undefined. The write value should be 0.	R
b23 to b16	BITMSK	Places comparative masks on a bit basis for Destination Address[7:0]. Bits [23:16] correspond to Destination Address[7:0]. Bits that are set to 0 are not subject to comparison. For example, if the bits BITMSK[2:0] of the mask register are set to 0, Destination Address[2:0] are not subject to comparison. That is, the frame is included if only the Destination Address [47:3] bits match.	R/W
b15 to b8	MADDR6B	Indicates the 6th byte from the beginning of the MAC address to be included.	R/W
b7 to b0	MADDR5B	Indicates the 5th byte from the beginning of the MAC address to be included.	R/W

3.4.2.13 GMAC_RXFIFO — RX FIFO Status Register

The GMAC_RXFIFO register is a status register that indicates the status of the receive FIFO.

Address: 400F 0200h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	RFULL	REMP	RRT	RSW												—
Value after reset	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Table 3.25 GMAC_RXFIFO Register Contents

Bit Position	Bit Name	Function	R/W
b31	RFULL	This bit is set to 1 when the number of empty words in RX FIFO is equal to or less than RFULLTH. (RFULLTH is set by using the GMAC_RXMODE register.)	R
b30	REMP	This bit is set to 1 when the number of words in RX FIFO is equal to or less than REMPTH. (REMPTH is set by using the GMAC_RXMODE register.)	R
b29	RRT	This bit is set to 1 when the number of words in RX FIFO buffer is over RRTTH. (RRTTH is set by using the GMAC_RXMODE register.)	R
b28 to b17	RSW	Indicates the number of data words in RX FIFO.	R
b16 to b0	Reserved	When read, the value returned is undefined. The write value should be 0.	R

3.4.2.14 GMAC_TXFIFO — TX FIFO Status Register

The GMAC_TXFIFO register is a status register that indicates the status of the transmit FIFO.

Address: 400F 0204h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TFULL	TEMP	TSTATUS			TRBFR			—	—	—	—	—	—	—	—
Value after reset	0	1	1	0	0	0	0	0	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Table 3.26 GMAC_TXFIFO Register Contents

Bit Position	Bit Name	Function	R/W
b31	TFULL	This bit is set to 1 when the empty space in the TX FIFO buffer is below the threshold set by the TFULLTH[1:0] bits of the GMAC_TXMODE register.	R
b30	TEMP	This bit is set to 1 when the number of data words in TX FIFO is equal to or less than the threshold value set by the TEMPTH[2:0] bits of the GMAC_TXMODE register.	R
b29 to b27	TSTATUS	Indicates the status of TX FIFO. The meanings of bit settings are as follows: 100b: ACC NEW FR: TX FIFO can accept new frames. 101b: WRITE ENABLE: TX FIFO can continue to accept frame data. 110b: CMPLT: Inclusion of one frame was completed. 111b: FULL: TX FIFO is in FIFO Full status. 000b to 011b: STOP: TX FIFO is inactive (or being initialized).	R
b26 to b24	TRBFR	Indicates the number of frames existing in the transmission result buffer.	R
b23 to b0	Reserved	These bits are read as 0.	R

3.4.2.15 GMAC_ACC — TCP/IPACC Register

The GMAC_ACC register is used to control operation of the TCP/IP accelerator.

Address: 400F 0208h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	RTCPIPACC	TTCPIPEN	RTCPIPEN
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	0	1	1

Table 3.27 GMAC_ACC Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b3	Reserved	These bits are read as 0. The write value should be 0.	R
b2	RTCPIPACC	1: RX TCPIPACC Off Disables checksum support for the RX TCPIP accelerator. Padding in the MAC header is inserted. 0: Checksum support for the RX TCPIP accelerator is enabled (initial value)	R/W
b1	TTCPIPEN	1: TX TCPIP Enable Enables the TX TCPIP accelerator. 0: TX TCPIP Disable Disables the TX TCPIP accelerator. The padding in the MAC header is also disabled, that is, there is no need to insert pad to the transmit frame.	R/W
b0	RTCPIPEN	1: RX TCPIP Enable Enables the RX TCPIP accelerator. 0: RX TCPIP Disable Disables the RX TCPIP accelerator. Padding in the MAC header section is not inserted.	R/W

3.4.2.16 GMAC_RXMAC_ENA — RX MAC Enable Register

The GMAC_RXMAC_ENA register is used to control operation of the receive MAC.

Address: 400F 0220h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RMACEN
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	1

Table 3.28 GMAC_RXMAC_ENA Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b1	Reserved	These bits are read as 0. The write value should be 0.	R
b0	RMACEN	1: Enables reception (initial value). 0: Disables reception.	R/W

3.4.2.17 BUFID — Receive Buffer Information Register

The BUFID register indicates information of the receive buffer (whether or not data exists, the address of the buffer holding received data, and the number of words of data). If the reception MACDMAC has completed data transfer, the receive buffer information is written to this register and held up to 64 pieces of information. If the receive buffer has data, the Ethernet MACDMA reception completion interrupt (HWRTOS_ETHDMAIR_Int) occurs. This interrupt stays active until the receive buffer becomes empty (i.e. the receive buffer information is read and the NOEMP bit becomes 0).

ADDR[15:0]:

The ADDR bits cannot indicate an address in the 32-bit address space. Therefore, to access a memory mapped buffer, use an offset of 0800 0000h.

To calculate the receive buffer Address(es):

1. Obtain the value of the ADDR bits.
2. Shift the value to the left by 11 bits.
3. Add an offset of 0800 0000h.

WORD[11:0]:

The number of words indicated by the WORD bits includes the number of words in the receive frame information. Therefore, the start address of the receive frame information is calculated as follows.

To calculate the start address of the receive frame information:

1. Obtain the value of the WORD bits.
2. Shift the value to the right by 16 bits.
3. Add the number of words shifted in step 2 to the receive buffer address as an offset.
4. Negatively offset the value by 2 words (the size of the receive frame information).

Address: 400F 1100h

Bit	b31	b30	b29	b28	b2711	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	NOEMP	—		VALID	WORD											
Value after reset	0	X	X	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ADDR															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 3.29 BUFID Register Contents

Bit Position	Bit Name	Function	R/W
b31	NOEMP	1: The receive buffer has data. 0: The receive buffer has no data.	R
b30, b29	Reserved	These bits are read as 0. The write value should be 0.	R
b28	VALID	1: The receive data is valid. 0: The receive data is invalid.	R
b27 to b16	WORD	Number Of Words In Receive Data (including receive MAC information)	R
b15 to b0	ADDR	Receive Buffer Address (bits 26 to 11)	R

3.5 Operation

3.5.1 Initialization

The initialization sequence in this section is an example for preparation of system environments for using HW-RTOS GMAC under configuration below.

HW-RTOS GMAC configuration of this example:

- HW-RTOS GMAC is connected to the management port of Advanced 5port Switch (A5PSW).

Figure 3.2 Initialization of HW-RTOS GMAC (Flowchart)

(1) Initial Settings of HW-RTOS GMAC

First of all, use HW-RTOS and HW-RTOS GMAC reset register (RTOSRST) of R-IN Engine Accessory Register to release reset state.

Execute procedures listed below to set up the hardware functions.

The procedure for setting up the HW-RTOS GMAC:

(1) Release a protect of RTOSRST register by RINSPCMD register of R-IN Engine Accessory Register.

(2) Release a reset of HW-RTOS and HW-RTOS GMAC module by RTOSRST register.

(3) Get a protect of RTOSRST register by RINSPCMD register.

(4) Set 0000 0003h in the C0TYPE register.

(5) Set 0000 0003h in the C0STAT register.

(6) Set 0000 8004h in the CMD register.

(7) Wait until 8000 0000h is read from the R0 register. Afterwards, dummy-read the R1 register.

(8) Set 8000 0000h in the GMAC_RESET register to reset HW-RTOS GMAC module.

CAUTION

When using the hardware OS accelerator function, the above settings (1) to (7) are not required since it is controlled by setting up the OS accelerator function.

After the completion of setup, make initial settings in the registers below.

- MAC address register
- Mode register
- TX MODE register
- RX MODE register

3.5.2 Hardware Functions (HWF)

A hardware function (HWF) is defined as a function for reducing the load on the CPU, such as a DMAC or Ethernet communications accelerator.

HWF consists of a combination of hardware modules which are divided by function, and an overall function is defined for the set of individual hardware modules.

The following three functions are defined as hardware functions:

- Buffer allocator
- MAC DMA controller
- Buffer RAM DMA controller

The figure below is a block diagram of these hardware functions in context. Solid lines in the figure indicate the flow of data, while broken lines indicate a command interface with the hardware function.

Figure 3.3 Block Diagram of the Hardware Functions

3.5.2.1 Processing Flow

(1) Flow of Processing for Issuing the Hardware Function Call

If you are using a hardware function, follow the flowchart below to issue the hardware function call.

Note 1. The interrupt control process is necessary only if HW-RTOS is used.

Figure 3.4 Flow of Processing for Issuing the Hardware Function Call

CAUTION

If the hardware real-time OS is prohibited from dispatching, issuing hardware function call is failed.
In this case, bits [15:0] of return value register R0 indicates FFE7h.

3.5.2.2 Buffer Allocator

(1) Functional Overview

The buffer allocator is a module for controlling the buffer RAM.

The buffer RAM is a communications buffer to improve throughput in Ethernet transfer. Although the buffer RAM has 128 Kbytes, an area of 128 Mbytes is used as the logical space for the dynamic securing and releasing of memory space by the buffer allocator.

To use the buffer RAM, secure the required area (hereafter “buffer”) beforehand, and then issue the hardware function calls provided for the buffer allocator. When CPU or MAC DMA controller writes to an area which has not been secured, an interrupt happens. When buffer RAM DMA controller has access to an area which has not been secured, two types of operation happen. The one is an interrupt and the other one is to return an exception code on R0 register. These operation depend on hardware function call.

To reuse a buffer after having secured it, the buffer must be released after it has been used.

The outline of the functions is as follows:

- A long buffer of up to 2048 bytes and short buffer of up to 512 bytes are available.
- When securing a buffer, the size is specified in bytes.
- When releasing a buffer, the size can be specified for the whole area or as the location of a byte (the part of the buffer from that address is released).

The segments which constitute a buffer are of 128 bytes each. The buffer allocator controls each of these 128 bytes segments, and connect these segments in response to hardware function calls to provide these as buffers. Addresses are seen as continuous across contiguous segments.

Figure 3.5 Method of Controlling a Buffer

(2) Initializing

For normal operation of the ECC circuit, it is necessary to initialize the buffer RAM.

RAM initialize:

ECC error correction cannot be enabled before initializing the buffer RAM.

After the initializing sequence below, set ECC_ENABLE bit of RAMEDC register to 1 to enable ECC error correction.

Initializing sequence for the buffer RAM:

1. Get the buffer of 2048 byte by HWFNC_LongBuffer_Get.
2. Read data from the long buffer area.
3. Write the read data to the same address.
4. Repeat 64 times from step 1 to step 3 to refresh syndrome value of ECC of 128 Kbyte area.
5. Release the buffer of 2048 byte by HWFNC_Buffer_Release.
6. Repeat 64 times step 5 to release all buffers.

(3) Buffer Control Operation

In this section, short and long buffers are collectively referred to as “buffers”. A short buffer has up to four segments and a long buffer has up to 16 segments.

Figure 3.6 Buffer Structure

(a) Acquisition of Buffers (HWFNC_ShortBuffer_Get, HWFNC_LongBuffer_Get)

Buffers can be acquired by issuing an HWFNC_ShortBuffer_Get or HWFNC_LongBuffer_Get hardware function call.

The size of the buffer is specified in bytes when calling these hardware functions. The number of bytes does not have to reach a segment boundary. The value returned is the address where the buffer starts.

The maximum numbers of short and long buffers that can be acquired are as listed in **Table 3.30**. Even if fewer short and long buffers are acquired than the maximum, acquisition will fail if the total size of buffers of both sizes exceeds the maximum size imposed by the 128 Kbytes of buffer RAM.

Table 3.30 Number of Buffers that can be Acquired

Buffer Type	Maximum Number of Buffers that can be Acquired	Remarks
Short buffer	128	Up to 512 segments (= 64 KB)
Long buffer	64	Up to 1024 segments (= 128 KB)

The address structure of buffers is shown below. When a buffer is acquired, the function returns the address range from 0C00 0000h to 0FFF FFFFh and 0800 0000h to 0BFF FFFFh for a long buffer and short buffer, respectively.

Figure 3.7 Buffer Structure

If a short buffer is acquired, bits [24:18] are given an SBID (short buffer ID), which is used as an identifier for the buffer. The buffer area is allocated with the offset field as 0 to indicate the address where the buffer starts.

If a long buffer is acquired, bits [23:18] are given an LLID (linked long buffer ID), which is used as an identifier for the buffer. The buffer area is allocated with the offset field as 0 to indicate the address where the buffer starts.

(b) Releasing a Buffer (HWFNC_Buffer_Release)

The whole area of an acquired buffer can be released by calling the HWFNC_Buffer_Release hardware function. When calling the hardware function, specify the address where the acquired buffer to be released starts.

(c) Releasing Part of a Buffer (HWFNC_Buffer_Return)

By calling the HWFNC_Buffer_Return hardware function, desired bytes can be released, starting from the location of a byte within the acquired buffer. This is provided for efficiency in using the space; for example, when a frame is received, another resource can use the area obtained by releasing the area following the end of the received frame data. When executing this system call, the addresses where the buffer and the space to be released start must be given as arguments.

(d) Testing Memory and Initializing Buffers

Since it is not allocated at the time of a reset, buffer RAM is neither writable nor readable in that situation. Accordingly, to test the memory, execute the HWFNC_LongBuffer_Get system call, etc., to secure the full capacity of the buffer RAM and make that memory available for access. This enables subsequent checking of the memory and initializing its contents.

(e) List of Hardware Function Calls

The table below lists the hardware function calls.

If an argument of a hardware function call is invalid, an invalid system call error code is returned in the return value register, R0.

Table 3.31 HWFNC_LongBuffer_Get

Name	HWFNC_LongBuffer_Get	
Function	Acquires a long buffer for use in the transmission and reception of frames. A buffer can be acquired with any size in bytes between 1 and 2048. Long buffers are mainly used to hold the data sections of frames. The address where the acquired buffer starts is returned in R1 as the value returned.	
Command register		
SYSC[15:0]	5000h	
Argument registers		
R4[15:0]	Buffer Length	Required buffer length. Unit: bytes. 1 to 2048
R4[23:16]	Reserved	Always 0
R4[31:24]	Unused	
R5[31:0]	Unused	
R6[31:0]	Unused	
R7[31:0]	Unused	
Return value registers		
R0[1:0]	Result	00b or 01b, and R0[29] = 1: Success 10b: Invalid system call 11b: The buffer is insufficient
R0[28:2]	Unused	All 0
R0[29]	Complete	0: Hardware function call not completed 1: Hardware function call completed
R0[31:30]	Unused	All 0
R1[31:0]	First logical address of the buffer	[31:27] 00001b [26:24] 100b [23:18] LLID [17: 0] 0

Table 3.32 HWFNC_ShortBuffer_Get

Name	HWFNC_ShortBuffer_Get	
Function	Acquires a short buffer for use in the transmission and reception of frames. A buffer can be acquired with any size in bytes between 1 and 512. Short buffers are mainly used to hold the header sections of frames, the data sections of ICMP and MAC management frames, etc. The address where the acquired buffer starts is returned in R1 as the value returned.	
Command register		
SYSC[15:0]	5006h	
Argument registers		
R4[15:0]	Buffer Length	Required buffer length. Unit: bytes. 1 to 512
R4[31:16]	Reserved	
R5[31:0]	Unused	
R6[31:0]	Unused	
R7[31:0]	Unused	
Return value registers		
R0[1:0]	Result	00b, 01b: Success 10b: Invalid system call 11b: The buffer is insufficient
R0[28:2]	Unused	All 0
R0[29]	Complete	0: Hardware function call not completed 1: Hardware function call completed
R0[31:30]	Unused	All 0
R1[31:0]	First logical address of the buffer	[31:27] 00001b [26:25] 00b [24:18] SBID [17: 0] 0

Table 3.33 HWFNC_Buffer_Release

Name	HWFNC_Buffer_Release	
Function	Releases an acquired long or short buffer.	
Command register		
SYSC[15:0]	5001h	
Argument registers		
R4[31:0]	Buffer Length	First logical address of the buffer to be released. The value is returned in R1 following a call of HWFNC_LongBuffer_Get or HWFNC_ShortBuffer_Get.
R5[31:0]	Unused	
R6[31:0]	Unused	
R7[31:0]	Unused	
Return value registers		
R0[1:0]	Result	00b, 01b: Success 10b: Invalid system call 11b: A buffer is not definable at the given address.
R0[28:2]	Unused	All 0
R0[29]	Complete	0: Hardware function call not completed 1: Hardware function call completed
R0[31:30]	Unused	All 0
R1[31:0]	Unused	All 0

Table 3.34 HWFNC_Buffer_Return

Name	HWFNC_Buffer_Return	
Function	Releases some of the latter half of an acquired short or long buffer. Specifying the location where the address range to be released starts leads to the release of the part of the buffer beginning at that address. The address can be set as any byte. This HWF is for the efficient use of buffer resources, for example when a received frame is short.	
Command register		
SYSC[15:0]	5002h	
Argument registers		
R4[31:0]	First logical address of the buffer	First logical address of the buffer to be released. The value is returned in R1 following a call of HWFNC_LongBuffer_Get or HWFNC_ShortBuffer_Get.
R5[31:0]	First logical address of a part of buffer to be released	First logical address of a part of buffer to be released (the part of buffer at addresses beginning from this address is released).
R6[31:0]	Unused	
R7[31:0]	Unused	
Return value registers		
R0[2:0]	Result	000b, 001b: Success 010b: Invalid system call 011b: A buffer address specified by R4 has not been defined. 100b: The part of the buffer at the address specified by R5 has already been released.
R0[28:3]	Unused	All 0
R0[29]	Complete	0: Hardware function call not completed 1: Hardware function call completed
R0[31:30]	Unused	All 0
R1[31:0]	Unused	All 0

3.5.2.3 MAC DMA Controller

(1) Functional Overview

The MAC DMA controller is used to transfer data between the buffer RAM and Ethernet MAC.

In transmission, the DMAC transfers data to be transmitted from the buffer RAM to the Ethernet MAC; in reception, the DMAC transfers data received by the Ethernet MAC to the buffer RAM.

This allows improved throughput for communications.

Figure below is a block diagram of the MACDMAC in context and the respective interrupt signals.

Figure 3.8 Block Diagram of the MACDMAC in Context and Interrupt Signals

(2) DMA for the Reception MAC

The figure below shows an outline of processing by the reception MACDMAC. A hardware function call (HWFNC_MACDMA_RX_Enable) must be issued to enable operation of the reception MACDMAC. The reception MACDMAC remains active until HWFNC_MACDMA_RX_Disable is issued.

While active, the reception MACDMAC constantly monitors the state of the MAC Rx FIFO. When the FIFO holds a received frame, the reception MACDMAC sends a request for the acquisition of a long buffer (2048 byte) to the buffer allocator. Once the long buffer has been acquired, the reception MACDMAC reads data from the MAC Rx FIFO and writes the data sequentially from the start of the acquired long buffer.

After the completion of the full transfer of one frame, the reception MACDMAC writes the number of received words (one word: 32 bits) and the first logical address of the buffer to the BUFID register as information on reception. The information written to the BUFID is described in section “Receive Buffer Information Register (BUFID)”.

The BUFID can be read by the CPU and is capable of holding up to 64 pieces of information.

Figure 3.9 Outline of Processing by the Reception MACDMAC

(a) Description of the Individual Functions of the MAC DMA Controller

- Partial release of buffer space
  The reception MACDMAC automatically releases an unused area that has no received data in the last buffer to have been acquired (buffer return function call). However, if the unused area is no larger than 128 bytes (one segment), buffer return does not proceed. Buffer return is a function call to release part of the secured buffer area and differs from the buffer release function call that releases the whole area of a secured buffer.

- Full release of the buffer
  If the following conditions are satisfied, the reception MACDMAC automatically releases the acquired buffer (calls the buffer release function).

  (1) The result of executing the function call for the buffer acquisition request was failure (the buffer has no unused area).

  (2) The result of analyzing the Rx frame information is that the received frame is invalidated by HWFNC_MACDMA_RX_Control.

  (3) HWFNC_MACDMA_RX_Disable is executed under the following condition:
  The number of received words is not greater than 4092 words

  In the above cases (1) and (2), all received frames are discarded and the buffer is released. In case (3), the received frames are not discarded (data resides in the MAC Rx FIFO) but only the release of the buffer is executed, after which the reception MACDMAC is immediately disabled. In any of cases (1), (2), and (3), the result of reception is not written to the BUFID.

- Generation of an error interrupt
  An error interrupt is issued in response to detection of the reception MACDMAC having failed to continue operation for reception for some reason or data not having been received correctly. The source of an error interrupt can be checked by executing the hardware function call HWFNC_MACDMA_RX_Errstat.
  For details, see “List of hardware function calls”.

- Generation of reception completed interrupts
  If the BUFID has information on the reception of one or more frames, the reception completed interrupt goes to its active level. The reception completed interrupt remains active as long as the BUFID register is not empty; that is, it has information on the reception of one or more frames.
  The reception completed interrupt is de-asserted when the BUFID is read and becomes empty.

- Judging whether a received frame is valid or invalid
  Judgment of whether a received frame is valid or invalid leads to an HWRTOS_ETHRFI_Int (valid frame reception) or HWRTOS_ETHRFE_Int (error frame reception) interrupt being issued.
  Each interrupt has more than one source and the generation of interrupts is enabled for all sources in the initial state.
  A specified source can be disabled by executing HWFNC_MACDMA_RX_Control. The frame which corresponds to the disabled source is discarded by full release of the buffer.

Figure 3.10 Conceptual Diagram of Judging Whether a Received Frame is Valid or Invalid

(b) Usage

- Procedure for Reading and Releasing Buffers
  A buffer which has received data must always be released after use. An example of the procedure is given below.

  Example of reading and releasing a buffer

  (1) Read the BUFID register.

  (2) Shift the bits [27:16] read from BUFID 16 bits to the right to obtain the number of received words.

  (3) The bits [15:0] read from the BUFID are bits [26:11] of the address where the acquired buffer starts.
  The individual bits of the address where the acquired buffer starts are configured as follows.

  [31:27]: 00001b
  [26:11]: Equivalent to the bits [15:0] in the BUFID
  [10: 0]: Always 0

  (4) After using the buffer, specify the start address as an argument and issue the buffer release function call to release the buffer.

- Procedure for processing in response to an error interrupt
  An example of the recommended procedure for processing in response to an error interrupt is given below.
  The value of R0[7:0] obtained by the HWFNC_MACDMA_RX_ERRSTAT function call is hereafter called bits [7:0] of the result of reading the error state.

  (1) Bit [3] of the result of reading the error state = 1
  A function call to forcibly end MACDMA Rx has been executed.

  (a) If bit [0] of the result of reading the error state = 1, proceed to step 3).

  (b) If bits [2:0] of the result of reading the error state have the value 4 or 0, the interrupt source is the forced termination of reception while it was in progress and this does not represent a problem. Since the received frames are all discarded and the information is not written to the BUFID, nothing is done,

so simply return to normal processing. The reception MAC FIFO may still have frame data that was received, but in such cases, the hardware automatically discards that data before the next round of reception starts.

(2) Bit [2] of the result of reading the error state = 1
The size of the frame is at least 4096 words.

(a) If bit [0] of the result of reading the error state = 1, proceed to step 3).

(b) Received data are all stored. The start address is obtained by reading the BUFID.

(c) Buffers that are no longer required are released according to the method given as example (above: “Procedure for reading and releasing buffers”).

(d) Returns to normal processing.

(3) Bit [0] of the result of reading the error state = 1
The remaining capacity of the buffer is insufficient.

(a) If bit [2] of the result of reading the error state = 1 (the size of the received frame is at least 4096 words) is satisfied at the same time, the buffer capacity is considered temporarily insufficient, so nothing is done.

(b) If the remaining capacity of the buffer is considered insufficient, the buffer is released to provide space.

(c) Return to normal processing. Note that received frames may have been lost during this period.

(c) List of Hardware Function Calls

The table below lists the hardware function calls.

If an argument of a hardware function call is invalid, an invalid system call error code is returned in the return value register, R0.

Access to an access prohibited area (an area other than the buffer RAM, etc.) while the hardware function call is running leads to the return of an exception code in the return value register R0.

Table 3.35 HWFNC_MACDMA_RX_Enable

Name	HWFNC_MACDMA_RX_Enable	
Function	Enables DMA for the reception MAC, that is, the transfer of data to the buffer memory from the FIFO. As long as the reception DMAC is enabled, transfer starts automatically whenever the FIFO within the MAC collects received frames. Since the DMAC executes Get Buffer at this time, the buffer memory is automatically acquired.	
Command register		
SYSC[15:0]	5101h	
Argument registers		
R4[31:0]	Unused	
R5[31:0]	Unused	
R6[31:0]	Unused	
R7[31:0]	Reserved	Always 0
Return value registers		
R0[0]	Result	0: Success 1: Invalid system call
R0[28:1]	Unused	All 0
R0[29]	Complete	0: Hardware function call not completed 1: Hardware function call completed
R0[31:30]	Unused	All 0
R1[31:0]	Unused	All 0

CAUTION

- If this hardware function is called while it is not disabled (this function call is already being executed) or this hardware function is called while a buffer return or release operation is in progress after reception has been suspended, the result is an invalid system call.
- The number of bytes to be transferred at a time is from 4 to 2048 bytes. Exceeding this range leads to the generation of an exception.

Table 3.36 HWFNC_MACDMA_RX_Disable

Name	HWFNC_MACDMA_RX_Disable	
Function	Disables DMA for the reception MAC. When forced reset is enabled, the data being received are discarded and information on reception is not stored in the BUFID register. At this time, the buffer is automatically released. When forced reset is disabled, the buffer is not automatically released.	
Command register		
SYSC[15:0]	5102h	
Argument registers		
R4[0]	Forced reset	0: This function is disabled while reception is in progress. 1: If the reception DMAC is enabled, it is disabled even if reception is in progress (the reception DMAC is forcibly reset). Nothing is done if the reception DMAC is already disabled.
R4[31:1]	Unused	
R5[31:0]	Unused	
R6[31:0]	Unused	
R7[31:0]	Reserved	Always 0
Return value registers		
R0[1:0]	Result	00b: Success 01b: Invalid system call (the buffer is in use or reception is suspended) 10b: The function cannot be disabled since reception is in progress. (only if forced reset is enabled) 11b: The function has already been disabled. (only if forced reset is enabled)
R0[28:2]	Unused	All 0
R0[29]	Complete	0: Hardware function call not completed 1: Hardware function call completed
R0[31:30]	Unused	All 0
R1[31:0]	Unused	All 0

Table 3.37 HWFNC_MACDMA_RX_Control

Name	HWFNC_MACDMA_RX_Control	
Function	Controls enabling or disabling of the interrupt source corresponding to bits [8:0] of the received frame information.	
Command register		
SYSC[15:0]	510Bh	
Argument registers		
R4[8:0]	Interrupt source	Controls enabling or disabling of the interrupt source corresponding to each bit. 0: Interrupts disabled 1: Interrupts enabled (initial value)
R4[31:9]	Unused	
R5[31:0]	Unused	
R6[31:0]	Unused	
R7[31:0]	Unused	
Return value registers		
R0[0]	Result	0: Success 1: Invalid system call
R0[28:1]	Unused	All 0
R0[29]	Complete	0: Hardware function call not completed 1: Hardware function call completed
R0[31:30]	Unused	All 0
R1[31:0]	Unused	All 0

Table 3.38 HWFNC_MACDMA_RX_Errstat

Name	HWFNC_MACDMA_RX_Errstat	
Function	Obtains error interrupt sources for the reception MACDMAC.	
Command register		
SYSC[15:0]	510Dh	
Argument registers		
R4[31:0]	Unused	
R5[31:0]	Unused	
R6[31:0]	Unused	
R7[31:0]	Unused	
Return value registers		
R0[3:0]	Result	[0]: Buffer Get fails [1]: Always 0 [2]: Rx data size is over 4096 words (16 KB) [3]: HWFNC_MACDMA_Rx_Disable was issued with forced reset was enabled.
R0[28:4]	Unused	All 0
R0[29]	Complete	0: Hardware function call not completed 1: Hardware function call completed
R0[31:30]	Unused	All 0
R1[31:0]	Unused	All 0

(3) DMA for the Transmission MAC

(a) Usage

The transmission MACDMAC uses descriptors. The descriptors are located in the buffer memory. That is, the software must acquire a buffer for the descriptor by issuing using a hardware function call before DMA can be set up. This buffer can be long or short. One buffer can hold multiple descriptors.

A transmission descriptor is shown in detail in figure below, Transmission Descriptor. Note that a descriptor must start on a 64-bit boundary. If it is not on a 64-bit boundary, the result of trying to use it is in an invalid system call.

A descriptor is formed in a succession of a 32-bit address and a 32-bit transfer byte count. Address FFFF FFFFh indicates the end of a descriptor. The address field of a descriptor indicates the transmission start address, and the byte count indicates the number of bytes to be transmitted from that address.

The DMAC reads the first pair of address and byte count in a descriptor, and then writes the specified data to transmission MAC FIFO. After that, the DMAC reads the next pair of address and byte count, and then writes the specified data to the transmission MAC FIFO. The DMAC continues this processing until it reads the end of the descriptor (FFFF FFFFh).

Source start addresses in the descriptor can be specified in units of bytes. The size of data to be transmitted can be specified in units of bytes. If the data writing point in the transmission FIFO is not at a word boundary, the DMAC automatically inserts padding.

The transmission MACDMAC starts when a function call HWFNC_MACDMA_TX_Start is issued. When this function call is issued, the start address of the transmit descriptor must be specified in the argument register R4.

Note that if the address field is not FFFF FFFFh and 0 is specified in the descriptor byte count field, the DMAC ignores the address field and does not perform transmission. In this case, DMA reads the next descriptor.

If the value of an address field is incorrect (for example, the address is outside the buffer area) or the number of transfer bytes is incorrect (for example, continued access causes a buffer area overflow), a MACDMA transmission error interrupt occurs.

Figure 3.11 Transmission Descriptor

(b) Automatic Release of the Buffer

If the release bit of the transmission descriptor is 0, no buffer is released.

If the release bit is 1, the transmission MACDMAC uses a buffer release function call to automatically release a buffer from the buffer area whose start address is indicated by the relevant descriptor after completion of transmission.

(c) Example of Operation

The figure below shows an example of operation for transmission by combining multiple buffers for use by the transmission MACDMAC.

Two independent buffers of buffer 1 and buffer 2 are combined for transmission by the transmission MACDMA by allocating transmission descriptors at the consecutive 64-bit boundary addresses.

The area labelled “Unused” means that the data end before the end of the segment (that is, it does not end at the 128-byte boundary).

In transfer, the start address of transmit data need not necessarily be the start of the buffer.

Figure 3.12 Example of Transmission as One Frame by Combining Multiple Buffers

(d) List of Hardware Function Calls

The table below lists the hardware function calls.

If an argument of a hardware function call is invalid, an invalid system call error code is returned in the return value register, R0.

When error happens during hardware function call operation, the MACDMA transmission error causes an interrupt.

Table 3.39 HWFNC_MACDMA_TX_Start

Name	HWFNC_MACDMA_TX_Start	
Function	Transfers data from the buffer memory to the FIFO for Ethernet MAC. The address where the transmission descriptor starts is set in R4. When transfer ends, an interrupt is generated. The number of bytes to be transferred at a time is from 1 to 2048 bytes.	
Command register		
SYSC[15:0]	5100h	
Argument registers		
R4[31:0]	Address of the descriptor	Address of the transmission descriptor
R5[31:0]	Unused	
R6[31:0]	Unused	
R7[31:0]	Reserved	Always 0
Return value registers		
R0[1:0]	Result	0: Success 1: Invalid system call
R0[28:2]	Unused	All 0
R0[29]	Complete	0: Hardware function call not completed 1: Hardware function call completed
R0[31:30]	Unused	All 0
R1[31:0]	Unused	All 0

Table 3.40 HWFNC_MACDMA_TX_Errstat

Name	HWFNC_MACDMA_TX_Errstat	
Function	Obtains error interrupt sources for the transmission MACDMAC	
Command register		
SYSC[15:0]	510Ch	
Argument registers		
R4[31:0]	Unused	
R5[31:0]	Unused	
R6[31:0]	Unused	
R7[31:0]	Unused	
Return value registers		
R0[1:0]	Result	[0]: Memory Access Violation [1]: Memory Access Timeout or Descriptor Error or Automatic release buffer Error
R0[28:2]	Unused	All 0
R0[29]	Complete	0: Hardware function call not completed 1: Hardware function call completed
R0[31:30]	Unused	All 0
R1[31:0]	Unused	All 0

3.5.2.4 Buffer RAM DMA Controller

(1) Functional Overview

The buffer RAM DMA controller transfers data between the buffer RAM and data RAM or the buffer RAM and buffer RAM. It is used to transfer data for transmission by the MACDMAC to the buffer and to transfer data received by the MACDMAC to the data RAM.

(2) DMA Transfer

Control of the buffer RAM DMA controller for each form of transfer is described below.

(a) Transfer between the Buffer RAM and the Data RAM

Calling the HWFNC_Direct_Memory_Transfer hardware function starts transfer between the buffer RAM and data RAM. After calling the function, confirm its completion by reading bit 29 of the R0 register. At this time, DMA transfer has been completed.

(b) Replacing Data in the Buffer RAM or Data RAM

By executing the hardware function HWFNC_Direct_Memory_Replace, an area in the buffer RAM or data RAM can be overwritten by a desired 32-bit data pattern.

The start and end of the area to be written must be on 128-bit boundaries so the amount of data written must be a multiple of 128 bits. After calling the function, confirm its completion by reading bit 29 of the R0 register. At this time, writing of the data pattern has been completed.

(c) Transfer from the Buffer RAM to the Buffer RAM

By executing the hardware function HWFNC_INTBUFF_DMA_Start or HWFNC_INTBUFF_DMA_Start (descriptor), data can be transferred from the buffer RAM to the buffer RAM. After calling the function, confirm its completion by reading bit 29 of the R0 register. However, DMA transfer has not been completed at this time. Check the completion of DMA transfer by means of the InterBuffer DMA transfer complete interrupt.

(d) List of Hardware Function Calls

The table below lists the hardware function calls.

If an argument of a hardware function call is invalid, an invalid system call error code is returned in the return value register, R0.

When error happens during hardware function call operation, HWFNC_Direct_Memory_Transfer or HWFNC_Direct_Memory_Replace returns exception code on the return value register R0, while HWFNC_INTBUFF_DMA_Start or HWFNC_INTBUFF_DMA_Start (Descriptor) causes an interrupt by exception.

Table 3.41 HWFNC_Direct_Memory_Transfer

Name	HWFNC_Direct_Memory_Transfer	
Function	Transfers data from the data RAM to the buffer RAM or from the buffer RAM to the data RAM. Data cannot be transferred from the buffer RAM to the buffer RAM. For transfer from the buffer RAM to the buffer RAM, use HWFNC_INTBUFF_DMA_Start (however, data transfer from the data RAM to the data RAM is possible).	
Command register		
SYSC[15:0]	5211h	
Argument registers		
R4[31:0]	Source address	Specifies the address where the source area for transfer starts.
R5[31:0]	Destination address	Specifies the address where the destination area for transfer starts.
R6[31:0]	Transfer size	Specifies the number of bytes for transfer.
R7[31:0]	Unused	
Return value registers		
R0[1:0]	Result	00b: Success 01b: Invalid system call (transfer from the buffer RAM to the buffer RAM has been specified) 10b: An exception has occurred
R0[28:2]	Unused	All 0
R0[29]	Complete	0: Hardware function call not completed 1: Hardware function call completed
R0[31:30]	Unused	All 0
R1[31:0]	Address where the exception occurred	When an exception has occurred, this is the address where it occurred. In other cases, all 0s.

Table 3.42 HWFNC_Direct_Memory_Replace

Name	HWFNC_Direct_Memory_Replace	
Function	Replaces the specified memory area in the data RAM or buffer RAM with a defined data pattern. The number of words to be written must be at least four.	
Command register		
SYSC[15:0]	5212h	
Argument registers		
R4[31:0]	Data Pattern	Specifies the data pattern for writing.
R5[31:0]	Start address	Specifies the address where the destination area for writing starts.
R6[31:0]	Number of words	Specifies the number of words to be written.
R7[31:0]	Unused	
Return value registers		
R0[1:0]	Result	00b: Success 01b: Invalid system call The set address was specified in byte units or the setting for the number of words to be transferred is three or fewer. 10b: An exception has occurred.
R0[28:2]	Unused	All 0
R0[29]	Complete	0: Hardware function call not completed 1: Hardware function call completed
R0[31:30]	Unused	All 0
R1[31:0]	Address where the exception occurred	When an exception has occurred, this is the address where it occurred. In other cases, all 0s.

Table 3.43 HWFNC_INTBUFF_DMA_Start

Name	HWFNC_INTBUFF_DMA_Start	
Function	Transfers data in the buffer memory. The source area for transfer starts at the address set in R4, the destination area for transfer starts at the address set in R5, and the number of bytes for transfer is set in R6. When transfer ends, an interrupt is generated.	
Command register		
SYSC[15:0]	5104h	
Argument registers		
R4[31:0]	Source address	Specifies the address where the source area for transfer starts.
R5[31:0]	Destination address	Specifies the address where the destination area for transfer starts.
R6[15:0]	Number of bytes for transfer	Specifies the number of bytes for transfer.
R6[31:16]	Unused	
R7[31:0]	Reserved	Always 0
Return value registers		
R0[0]	Result	0: Success 1: Invalid system call
R0[28:1]	Unused	All 0
R0[29]	Complete	0: Hardware function call not completed 1: Hardware function call completed
R0[31:30]	Unused	All 0
R1[31:0]	Unused	All 0

Table 3.44 HWFNC_INTBUFF_DMA_Start (Descriptor)

Name	HWFNC_INTBUFF_DMA_Start (Descriptor)	
Function	Transfers data in the buffer memory. When transfer ends, an interrupt is generated. This function requires a descriptor instead of an address and size as an argument.	
Command register		
SYSC[15:0]	5114h	
Argument registers		
R4[31:0]	Address where the transfer source descriptor starts	Specifies the address where the transfer source descriptor starts.
R5[31:0]	Address where the transfer destination descriptor starts	Specifies the address where the transfer destination descriptor starts
R6[31:0]	Unused	
R7[31:0]	Reserved	Always 0
Return value registers		
R0[0]	Result	0: Success 1: Invalid system call
R0[28:1]	Unused	All 0
R0[29]	Complete	0: Hardware function call not completed 1: Hardware function call completed
R0[31:30]	Unused	All 0
R1[31:0]	Unused	All 0

CAUTION

- The structure of the descriptor is the same as for the MACDMAC, but the function does not automatically release the buffer.
- If transfer size written in the source descriptor is differ to the one in the destination descriptor, the source descriptor side is high priority. When the sizes specified for the source and destination differ, the operation is as follows:
  - The transfer size in the source descriptor < The transfer size in the destination descriptor → The size in source descriptor is used with no problem.
  - The transfer size in the source descriptor > The transfer size in the destination descriptor → An exception may occur.

3.5.3 Interrupts

The interrupts that the Ethernet MAC generates are described below.

Table 3.45 Interrupts related to Operations for Transmission

Interrupt Name	Symbol	Conditions for Asserting and De-asserting Interrupts
TX FIFO underflow	HWRTOS_ETHTFIU_Int	This interrupt is generated when the transmission size specified in the descriptor and transmission frame control information are different. At this time, transmission does not proceed. Retransmit after fix the settings of the descriptor or the transmission frame information. Since this interrupt is generated as a pulse, de-asserting the interrupt source is not required
TX FIFO error interrupt	HWRTOS_ETHTFIE_Int	This interrupt is generated when information is further updated while the GMAC_TXID/GMAC_TXRESULT register is holding the maximum number of items of information (four). Take care that the oldest of the retained information will have been overwritten when this error occurs. Reading the GMAC_TXID/GMAC_TXRESULT register until the value of the GMAC_TXFIFO.TRBFR bit becomes 0 leads to clearing of the retained information and restoring normal operation.
MACDMA transmission error	HWRTOS_ETHDTIE_Int	This interrupt is generated when transmission MACDMA faces an error. There are several factors for the error, the error information is provided by HWFNC_MACDMA_TX_Errstat. Retransmit after fix the settings of the transmission descriptor. Since this interrupt is generated as a pulse, de-asserting the interrupt source is not required.
MACDMA transmission complete	HWRTOS_ETHDMAIT_Int	This interrupt is generated when DMA transfer from the buffer RAM to the transmission MAC FIFO is completed. At this time, DMA transfer has been completed but Tx MAC transmission has not been completed yet. Since this interrupt is generated as a pulse, de-asserting the interrupt source is not required.
transmission complete interrupt	HWRTOS_ETHIT_Int	This interrupt occurs when operations for communications by the transmission MAC are completed. Since this interrupt is generated as a pulse, de-asserting the interrupt source is not required.

Table 3.46 Interrupts related to Operations for Reception

Interrupt Name	Symbol	Conditions for Asserting and De-asserting Interrupts
MACDMA reception completion	HWRTOS_ETHDMAIR_Int	This interrupt is generated when MACDMA reception is successfully completed. It remains active until the BUFID register becomes empty of information on reception. The interrupt source is de-asserted when the BUFID is read and it becomes empty.
MACDMA reception error	HWRTOS_ETHDRIE_Int	This interrupt indicates that an error has occurred while the reception MACDMAC was operating. Because the interrupt may indicate more than one errors, HWFNC_MACDMA_RX_Errstat is used to get precise error factor. Since this interrupt is generated as a pulse, de-asserting the interrupt source is not required.
MACDMA valid frame reception completion	HWRTOS_ETHRFI_Int	This interrupt is generated when operations by the reception MACDMAC end normally and the received frame is valid. The interrupt source can be specified by referring to information on the received frame. It remains active until the BUFID register becomes empty of information on reception. The interrupt source is de-asserted when the BUFID is read and becomes empty.
MACDMA error frame reception completion	HWRTOS_ETHRFE_Int	This interrupt is generated when operations by the reception MACDMAC end normally and the received frame has an error. The interrupt source can be specified by referring to information on the received frame. It remains active until the BUFID register becomes empty of information on reception. The interrupt source is de-asserted when the BUFID is read and becomes empty.
RX FIFO overflow	HWRTOS_ETHRFIV_Int	This interrupt is generated when data are received while the buffer does not have enough space. When this error occurs, received data may already have been discarded. To restore to the normal state, release buffers. Since this interrupt is generated as a pulse, de-asserting the interrupt source is not required.

Table 3.47 Interrupts related to Other Operations

Interrupt Name	Symbol	Conditions for Asserting and De-asserting Interrupts
MII management access completion interrupt	HWRTOS_ETHMMAI_Int	This interrupt is generated when reading from or writing to the MII management bus is completed. Since this interrupt is generated as a pulse, de-asserting the interrupt source is not required.
Ether pause packet transmission completion	HWRTOS_ETHPPIT_Int	This interrupt is generated when the transmission of a pause packet is completed. Since this interrupt is generated as a pulse, de-asserting the interrupt source is not required.
InterBuffer DMA transfer completion	HWRTOS_BUFDMA_Int	This interrupt is generated if DMA transfer between buffer RAMs is completed. Since this interrupt is generated as a pulse, de-asserting the interrupt source is not required.
InterBuffer DMA error	HWRTOS_BUFDMAERR_Int	This interrupt is generated if DMA access reaches to unassigned buffer area during transfer between buffer RAMs. Since this interrupt is generated as a pulse, de-asserting the interrupt source is not required.
Buffer RAM area access error	HWRTOS_BRAMERR_Int	This interrupt is generated if CPU have access to the buffer area which is not acquired. Since this interrupt is generated as a pulse, de-asserting the interrupt source is not required.

3.5.4 Ethernet Frame Transmission Function

This section describes transmission of Ethernet frames. For Ethernet MAC, Ethernet frames are transmitted in the following sequence:

(9) Initial setup. (see **Section 3.5.1, Initialization**)

(10) Get a transmit buffer.

(11) Create Transmit Frame Control Information.

(12) Create Ethernet frame data.

(13) Create Transmit descriptors.

(14) DMA starts by executing HWFNC_MACDMA_TX_Start command.

(15) DMA transfer from buffer RAM to TX FIFO is performed according to transmit descriptors.

(16) Ethernet transmission is started by MAC according to the transmit frame control information included in the transmit data.

(17) A transmission completion interrupt occurs.

(18) Post transmission processing, including the status check, is performed.

(19) Release the transmit buffer. (Optional)

The above steps are described in the following sections.

3.5.4.1 Get a Transmit Buffer

Get a transmit buffer by HWFNC_LongBuffer_Get call.

3.5.4.2 Create Transmit Data

Tx frame control information format is shown as below. A transmission descriptor points the start address of this frame.

In Gigabit Ethernet MAC (HW-RTOS GMAC), a transmission frame size and all kinds control are dictated by 64 bits Tx frame control information added before normal Ethernet frame data.

Figure 3.13 TX Data Format

CAUTION

Make sure that the TX data conforms to this format.

(1) Transmit Frame Control Information

When RMTAGCTRL.MGMT_ENB = 1, Tx frame control information is shown below.

Table 3.48 Transmit Frame Control Information Format (RMTAGCTRL.MGMT_ENB = 1)

	31	30	…	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
TX Frame Control Information		TX_WORD[12:0]			TX_EOB[1:0]		Use provided sequence number for the PRP	Force PRP trailer append	Suppress PRP trailer append	One-step correction field update	Transmit Timestamp	Reserved (always 0)	Forced Forwarding, include filtering	Forced Forwarding	Reserved (always 0)		TCPIP ACC OFF	ITAG	ICRC	APAD	Reserved (always 0)		0
	Control Data2[31:16] (SEQ ID for RCT)						Reserved				Control Data2 [3:0] (Port[3:0])				Frame ID[7:0]								1

When RMTAGCTRL.MGMT_ENB = 0, Tx frame control information is shown below.

Table 3.49 Transmit Frame Control Information Format (RMTAGCTRL.MGMT_ENB = 0)

	31	30	…	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
TX Frame Control Information		TX_WORD[12:0]			TX_EOB[1:0]		Reserved (always 0)										TCPIP ACC OFF	ITAG	ICRC	APAD	Reserved (always 0)		0
	Frame ID[31:0]																						1

The fields of the transmit frame control information are described below.

Table 3.50 Transmit Frame Control Information

Field Name	Description
TX_WORD[12:0]	The number of words of transmission Ethernet frame. The number of valid bytes in last word is dictated by TX_EOB [1:0].
TX_EOB[1:0]	Valid octet number in the last word in this frame. 00b: 1 byte valid 01b: 2 bytes valid 10b: 3 bytes valid 11b: 4 bytes valid
SEQ ID for RCT*1	When Use provided sequence number for the PRP trailer bit is 1, the 16-bit sequence number for use with the frame when appending the redundancy control trailer (RCT). reserved otherwise.
Port [3]*1	The Forced Forwarding function of the Ethernet switch is permitted for Port 3.
Port [2]*1	The Forced Forwarding function of the Ethernet switch is permitted for Port 2.
Port [1]*1	The Forced Forwarding function of the Ethernet switch is permitted for Port 1.
Port [0]*1	The Forced Forwarding function of the Ethernet switch is permitted for Port 0.
Use provided sequence number for the PRP trailer*1	Indicates that the sequence number for the frame to be used when appending the RCT is found in SEQ ID for RCT. The bit has an effect only if Force PRP trailer append = 1. Otherwise, this bit is ignored.
Force PRP trailer append*1	Indicates that the transmitter must append a redundancy control trailer (RCT) when sending the frame at a port defined in the PRP group. The bit has an effect only if Suppress PRP trailer append = 0. Otherwise, this bit is ignored.
Suppress PRP trailer append*1	Suppresses the switch adding a redundancy control trailer (RCT) when sending the frame at a port defined in the PRP group (i.e. bypassing the RedBox function). The bit has no effect if transmission occurs at a normal port.
one-step correction field update*1	When enabled, the correction field of the frame is subtracted from the tx timestamp upon transmission and the result replaces the correction field.
Transmit Timestamp*1	The timestamp function for transmit frames is enabled when an Ethernet switch is used.
Force Forwarding, include filtering*1	If set together with forced forwarding, normal filtering of the destination port mask applies (i.e. disabled ports will be removed from the list). If 0, the frame is forwarded also to disabled ports. ***Note)*** This applies to BPDU frames only. Normal frames will be filtered always (i.e. can never be transmitted to disabled ports).
Forced Forwarding*1	The Forced Forwarding function of the Ethernet switch is enabled. If this function is enabled, frames are output from the specified port regardless of the filter settings of the switch.
TCPIP ACC OFF	1: Disable TCPIP accelerator 0: Enable TCPIP accelerator
ITAG	Indicates that this frame has VLAN Tag
ICRC	Indicates that the frame written to FIFO has already included CRC The APAD field is ignored if this bit is set.
APAD	Indicates that the frame is automatically padded if its length is shorter than 64 octets.
Frame ID[31:0]	Specifies a frame identifier.

Note 1. These functions are only available when insertion of a management tag is permitted by Ethernet switch management TAG control register (RMTAGCTRL and MGMT_TAG_CONFIG). If it's not permitted to insert a management tag, these fields are not effective.

In cases where TX_WORD [12:0] and TX_EOB [1:0] are combined into TX_LENGH [14:0] (15 bits), TX_LENGH [14:0] can be calculated from the following formula based on the Ethernet frame size (in bytes):
TCPIPACC Pad Size is 2 when Tx TCPIPACC is enabled (GMAC_ACC.TTCPIPEN = 1) and 0 when it is disabled.

$$\text{TX_LENGH}\,[14:0] = (\text{TX frame size} - \text{TCPIPACC Pad Size} + 3)(\text{bytes})$$

(2) Ethernet Frame

The transmission Ethernet frame data format and the description of the fields are given below.

Table 3.51 Transmit Ethernet Frame

Field Name	Description
Destination MAC Address	MAC address of destination
Source MAC Address	MAC address of source
Type / Length	Ethernet Type or Length
TPID	Tag Protocol Identifier. This field is available If VLAN Tag is included.
VLAN Info	Tag Control Information. This field is available If VLAN Tag is included.
Frame Payload	Payload

(a) When Tx TCPIP Accelerator is Enabled

If the Tx TCPIP accelerator function is enabled (GMAC_ACC.TTCPIPEN = 1), Ethernet frame data requires 2-byte padding between the Type/Length field and Payload.

Table 3.52 Tx Ethernet Frame Data Format — TCPIPACC is Enabled, without VLAN Tag

	31	30	…	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Ethernet Frame Data	Destination MAC Address (6 bytes)																						2
							Source MAC Address (6 bytes)																3
																							4
	Padding (2 bytes)						Type / Length (2 bytes)																5
	Frame Payload (Max. 1500 bytes)																						6 : : N

Table 3.53 Tx Ethernet Frame Data Format — TCPIPACC is Enabled, with VLAN Tag

	31	30	…	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Ethernet Frame Data	Destination MAC Address (6 bytes)																						2
							Source MAC Address (6 bytes)																3
																							4
	Padding (2 bytes)						TPID (2 bytes)																5
	Type / Length (2 bytes)						VLAN Info (2 bytes)																6
	Frame Payload (Max. 1500 bytes)																						7 : : N

CAUTION

Padding (2 bytes) can be any value.

Padding (2 bytes) is not included in the specified size of Ethernet frames (TX_WORD[12:0], TX_EOB[1:0]).

(b) Tx TCPIP Accelerator is Disabled

The Ethernet frame data formats when the Tx TCPIP accelerator function is disabled (GMAC_ACC.TTCPIPEN = 0) are shown below.

Table 3.54 Tx Ethernet Frame Data Format — TCPIPACC is Disabled, without VLAN Tag

	31	30	…	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Ethernet Frame Data	Destination MAC Address (6 bytes)																						2
							Destination MAC Address (6 bytes)																3
	Source MAC Address (6 bytes)																						4
	Source MAC Address (6 bytes)						Type / Length (2 bytes)																5
	Frame Payload (Max. 1500 bytes)																						6 : : N

Table 3.55 Tx Ethernet Frame Data Format — TCPIPACC is Disabled, with VLAN Tag

	31	30	…	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Ethernet Frame Data	Destination MAC Address (6 bytes)																						2
							Destination MAC Address (6 bytes)																3
	Source MAC Address (6 bytes)																						4
	VLAN Info (2 bytes)						TPID (2 bytes)																5
	VLAN Info (2 bytes)						Type / Length (2 bytes)																6
	Frame Payload (Max. 1500 bytes)																						7 : : N

3.5.4.3 Create Transmit Descriptors

Descriptors which DMA controller for transmission MAC uses are as follows.

After making descriptors, transmission operation begins by activating transmission DMA.

Figure 3.14 TX Descriptor Format

The start address of a descriptor must be at a 64-bit boundary ([2:0] = 0). If it is not at a 64-bit boundary, an error will be stored in return value register R0.

A descriptor is formed in a succession of a 32-bit address and a 32-bit transfer byte count. Address FFFF FFFFh indicates the end of a descriptor. The address field of a descriptor indicates the transmission start address, and the byte count indicates the number of bytes to be transmitted from that address. The DMAC reads the first pair of address and byte count in a descriptor, and then writes the specified data to transmission MAC FIFO. After that, the DMAC reads the next pair of address and byte count, and then writes the specified data to the transmission MAC FIFO. The DMAC continues this processing until it reads the end of the descriptor (FFFF FFFFh).

Source start addresses in the descriptor can be specified in units of bytes. The size of data to be transmitted can be specified in units of bytes. If the data writing point in the transmission FIFO is not at a word boundary, the DMAC automatically inserts padding.

The transmission MACDMAC starts when “start of transmission” is issued as a hardware function call. When this function call is issued, the start address of the transmit descriptor must be specified in the R4 register.

Note that if the address field is not FFFF FFFFh and 0 is specified in the descriptor byte count field (14 bits), the DMAC ignores the address field and does not perform transmission. In this case, the DMAC reads the next descriptor.

If the value of an address field is incorrect (for example, the address is outside the buffer area) or the number of transfer bytes is incorrect (for example, continued access causes a buffer area overflow), an error interrupt occurs.

If Release Bit is 1, the transmission MACDMAC uses a buffer release function call to automatically release a buffer from the buffer area whose start address is indicated by the relevant descriptor after completion of transmission. If Release Bit is 0, no buffer is released.

3.5.4.4 Transmission Start

Start to transmit by HWFNC_MACDMA_TX_Start call.

3.5.4.5 Completing Transmission

The Ethernet MACDMA transmission complete interrupt occurs when DMA transfer has been completed, and the Ethernet transmission complete interrupt occurs when MAC transmission has been completed.

If the TX buffer which is already acquired is to be reused for the next transmission, acquisition of the TX buffer is not required.

3.5.5 Ethernet Frame Reception Function

This section describes reception of Ethernet frames. For Ethernet MAC, Ethernet frames are received in the following sequence:

(1) Initial setup. (see **Section 3.5.1, Initialization**)

(2) Enable reception MAC.

(3) Enable reception DMA.

(4) Receive a frame and get a buffer.

(5) Reception completion interrupt occurs.

(6) Get Receive Buffer Information.

(7) Check frame status in Receive Frame Information.

(8) Get Ethernet frame data.

(9) Release the receive buffer.

3.5.5.1 Enable Reception MAC

Enable reception MAC by setting 1 to reception permission register (GMAC_RXMAC_ENA).

3.5.5.2 Enable Reception DMA

Enable reception DMA by HWFNC_MACDMA_RX_Enable call.

3.5.5.3 Receive a Frame and Get a Buffer

When frame reception occurs, reception DMA controller get a receive buffer by invoking HWFNC_LongBuffer_Get call automatically.

3.5.5.4 Get Receive Buffer Information

After a reception completion interrupt occurs, read Receive Buffer Information register (BUFID) to get an address and size of the buffer for receive data.

Referring the address, get a receive frame information and Ethernet frame data.

Please refer to next chapter for the format of the reception data.

3.5.5.5 Format of Receive Data

In the reception of frames by the gigabit Ethernet MAC, 64 bits of received frame information will be appended after the frame data. This information indicates the state of reception: Size of the Ethernet frame, errors, etc.

Since the received frame information starts on a 64-bit boundary, the amount of padding following the Ethernet frame varies with the frame size.

Figure 3.15 Rx Data Format

(1) Receive Frame Information

The allocation and descriptions of the fields of the received frame information are given below.

Table 3.56 Receive Frame Information

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Rx Frame Information (Word Address: N to (N+2))	Reserved (always 0)	RX_WORD[12:0]													RX_EOB[1:0]		Reserved (always 0)	Reserved (always 0)	MAR[4:0]					MARBR	MARMLT	MARIND	MARNOTMT	TOOLONG	TOOSHORT	FIFOOVF	NBLERR	CRCERR
	SESSION_ID[15:0]																Reserved (always 0)	Reserved (always 0)	Reserved (always 0)	Reserved (always 0)	Reserved (always 0)	MARSTAT[2:0]		IPNG	TCPNG	IPV6NG	OUT_OF_LIST	TYPEIP	MAACL	PPPOE	VTAG

Table 3.57 Receive Frame Information (1/2)

Field name	Description
SESSION_ID[15:0]	If this field is set to “1”, it indicates Session ID of PPPoE Session Stage.
MARSTAT[2:0]	MARSTAT[2]: If this field is set to “1”, it indicates Broadcast Address. MARSTAT[1]: If this field is set to “1”, it indicates Multicast Address. MARSTAT[0]: If this field is set to “1”, it indicates Individual Address.
IPNG*2	If this field is set to “1”, it indicates that the checksum of the IPv4 header conflicts with the calculation result of the TCPIP accelerator.
TCPNG*2	If this field is set to “1”, it indicates that the checksum of the TCP or UDP header conflicts with the calculation result of the TCPIP accelerator.
IPV6NG*2	If this field is set to “1”, it indicates that analysis of the IPv6 extended header is Routing, Hop-by-Hop, or Destination Opt, and also the header length field is invalid.
OUT_OF_LIST*2	If this field is set to “1” when IPv6 is used, it indicates that a protocol number not listed below was detected in the expansion header. 0x06 (TCP header) 0x11 (UDP header) 0x00 (Hop-by-Hop) 0x3C (Destination Opt) 0x2C (Fragment) 0x2B (Routing) 0x3B (No next header) 0x32 (ESP header) 0x33 (AH header)
TYPEIP*2	If this field is set to “1”, it indicates that an IP packet was received.
MAACL*2	If this field is set to “1”, it indicates that an 802.3 (LLC/SNAP) packet was received.
PPPOE*2	If this field is set to “1”, it indicates that a PPPoE packet was received.
VTAG*2	If this field is set to “1”, it indicates that a packet with VTAG was received.
RX_WORD[12:0]	Number of Ethernet Frame words
RX_EOB[1:0]	Indicates the number of valid bytes in the last word of this frame.*1 00b: 1 byte valid 01b: 2 bytes valid 10b: 3 bytes valid 11b: 4 bytes valid

Table 3.57 Receive Frame Information (2/2)

Field name	Description
MAR[4:0]	MAR[4:1]: Unused (Fixed 0) MAR[0]: Indicates that the Pause Packet destination address was received.
MARBR	If this field is set to “1”, it indicates that the receive frame is a Broadcast address.
MARMLT	If this field is set to “1”, it indicates that the receive frame is a Multicast address.
MARIND	If this field is set to “1”, it indicates that the receive frame is a packet from an address registered in the MAC address register.
MARNOTMT	If the receive frame is not an address for this station, this field is set to “1”.
TOOLONG	If this field is set to “1”, it indicates that the receive frame is longer than the maximum frame length (1518 octets). If insertion of management tags is permitted, the tag (8 bytes) is inserted to Ethernet frame. Therefore, in this case, this field is set to “1” if receive frame is longer than 1510.
TOOSHORT	If this field is set to “1”, it indicates that the receive frame is shorter than the minimum frame length (64 octets). This MAC never receives a packet in which TOOSHORT is “1” because the TOOSHORT packet is automatically discarded.
FIFOOVF	If this field is set to “1”, it indicates that the RX FIFO buffer overflows during frame reception. When this bit is set, received data may be invalid.
NBLERR	If this field is set to “1”, it indicates that the words in the receive frame have an error such as an encoding error.
CRCERR	If this field is set to “1”, it indicates that the receive frame has a CRC error.

Note 1. The FCS of an Ethernet frame (4 bytes) and padding of the MAC header to be inserted by the Rx TCPIP accelerator function (2 bytes) are also included in the number of received bytes.

Note 2. If Rx TCPIP accelerator is disabled, these fields are invalid.

If RX_WORD[12:0] and RX_EOB[1:0] are combined into RX_LENGTH [14:0] (by using RX_WORD[12:0] as the upper bits and RX_EOB[1:0] as the lower bits), the number of bytes in the received frame can be calculated by using the following expression:

$$(\text{Number of receive bytes in the Ethernet frame}) = \text{RX_LENGTH}\ [14:0] - 3$$

Examples:

- If Rx data is 1 byte → RX_WORD = 0x1 RX_EOB = 0x0 → 4 − 3 = 1 (byte)
- If Rx data is 8 bytes → RX_WORD = 0x2 RX_EOB = 0x3 → 11 − 3 = 8 (bytes)
- If Rx data is 5 bytes → RX_WORD = 0x2 RX_EOB = 0x0 → 8 − 3 = 5 (bytes)
- If Rx data is 9 bytes → RX_WORD = 0x3 RX_EOB = 0x0 → 12 − 3 = 9 (bytes)

(2) Ethernet Frame

The data format of received Ethernet frames is shown below.

Table 3.58 Data Format of Received Ethernet Frames

Item	Explanation
Destination MAC Address	MAC Address of Destination If insertion of management tags is permitted by the Ethernet switch management TAG control register (RMTAGCTRL), management TAG information is stored.
Source MAC Address	MAC Address of Source
TPID	Tag Protocol Identifier. This field is available If VLAN Tag is included.
VLAN Info	Tag Control Information. This field is available If VLAN Tag is included.
Type / Length	Ethernet Type or Length
Frame Payload	Payload
FCS	Frame Check Sequence If the Rx TCPIP accelerator function is enabled and the received packet has TCP/UDP, the FCS field is overwritten by the TCP/UDP checksum. This checksum can be used to calculate the total checksum of fragmented TCP/UDP packets.

(a) If Insertion of Management Tags is Permitted

If insertion of management tags is permitted by the Ethernet switch management TAG control register (RMTAGCTRL), the Destination MAC Address [47:0] field is used as follows:

Table 3.59 Destination MAC Address Field (when insertion of management tag is permitted)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Ethernet Frame	Time Stamp[31:0]																															
	Source Address[15:0]																Destination Address[7:0]								Port Number[3:0]				MAC Add Entry[3:0]			

Table 3.60 Destination MAC Address Field Description

Item	Explanation
Time Stamp [31:0]	Received timestamp of the frame.
MAC Add Entry [3:0]	The index numbers of MAC address registers (GMAC_ADR[n+1]A, GMAC_ADR[n+1]B, n = 0 to 15) which correspond to the received frame. ***Example:*** value = 5 A destination address of a frame corresponds to setting of GMAC_ADR6A and GMAC_ADR6B.
Port Number[3:0]	Port number where the frame was received from.
Destination MAC Address	MAC address of destination
Source MAC Address	MAC address of source

CAUTION

If the AFILLTEREN bit of the GMAC_RXMODE register is set to 1, it is impossible to recover the destination MAC address because the MAC Add Entry field is invalid.

(b) When Rx TCPIP Accelerator is Enabled and a Frame has No TCP/UDP Packet

Table 3.61 Format of Receive Ethernet Frame — TCPIPACC is Enabled, without VLAN Tag, No TCP/UDP Packets

Ethernet Frame (Word Address: 0 to N)	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Destination MAC Address (6 bytes)																															
	Source MAC Address (6 bytes)																															
	Padding (2 bytes)																Type / Length (2 bytes)															
	Payload (Max: 1500 bytes) & FCS (4 bytes)																															
	Padding (0 to 7 bytes)																															

Table 3.62 Format of Receive Ethernet Frame — TCPIPACC is Enabled, with VLAN Tag, No TCP/UDP Packets

Ethernet Frame (Word Address: 0 to N)	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Destination MAC Address (6 bytes)																															
	Source MAC Address (6 bytes)																															
	Padding (2 bytes)																TPID (2 bytes)															
	Type / Length (2 bytes)																VLAN Info (2 bytes)															
	Payload (Max: 1500 bytes) & FCS (4 bytes)																															
	Padding (0 to 7 bytes)																															

(c) When Rx TCPIP Accelerator is Enabled and a Frame has TCP/UDP Packets

Table 3.63 Format of Receive Ethernet Frame — TCPIPACC is Enabled, without VLAN Tag, with TCP/UDP Packets

Ethernet Frame (Word Address: 0 to N)	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Destination MAC Address (6 bytes)																															
																	Source MAC Address (6 bytes)															
	Padding (2 bytes)																Type / Length (2 bytes)															
	Payload (Max: 1500 bytes)																															
	Padding (0 to 3 bytes)																															
																	TCP/UDP Check Sum (2 bytes)															
	Padding (2 or 6 bytes)																															

Table 3.64 Format of Receive Ethernet Frame — TCPIPACC is Enabled, with VLAN Tag, with TCP/UDP Packets

Ethernet Frame (Word Address: 0 to N)	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Destination MAC Address (6 bytes)																															
																	Source MAC Address (6 bytes)															
	Padding (2 bytes)																TPID (2 bytes)															
	Type / Length (2 bytes)																VLAN Info (2 bytes)															
	Payload (Max: 1500 bytes)																															
	Padding (0 to 3 bytes)																															
																	TCP/UDP Check Sum (2 bytes)															
	Padding (2 or 6 bytes)																															

(d) When Rx TCPIP Accelerator is Disabled

Table 3.65 Format of Receive Ethernet Frame — TCPIPACC is Disabled, without VLAN Tag

Ethernet Frame (Word Address: 0 to N)	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Destination MAC Address (6 bytes)																															
																	Source MAC Address (6 bytes)															
																	Type / Length (2 bytes)															
	Payload (Max: 1500 bytes) & FCS (4 bytes)																															
																	Padding (0 to 7 bytes)															

Table 3.66 Format of Receive Ethernet Frame — TCPIPACC is Disabled, with VLAN Tag

Ethernet Frame (Word Address: 0 to N)	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Destination MAC Address (6 bytes)																															
																	Source MAC Address (6 bytes)															
	VLAN Info (2 bytes)																TPID (2 bytes)															
																	Type / Length (2 bytes)															
	Payload (Max: 1500 bytes) & FCS (4 bytes)																															
																	Padding (0 to 7 bytes)															

3.5.6 TCPIP Accelerator Function

If the TCPIP accelerator function is enabled, hardware can calculate the checksum for transmission or reception of packets. The following three protocols are targets for checksum calculation.

- IPv4 header checksum
- TCP checksum
- UDP checksum

This section explains how to use the TCPIP accelerator for transmission and reception.

(1) Transmission Using the TCPIP Accelerator

When the TTCPIPEN bit of the GMAC_ACC register is set to 1, the TCPIP accelerator for transmission is enabled. If a packet including IPv4, TCP/IP, or UDP/IP is transmitted while the TCPIP accelerator is enabled, hardware automatically calculates the checksum and writes it to the checksum field in the packet. The TCPIP accelerator requires 2-byte padding in the MAC header.

In addition, if the TCPIP ACC OFF field of Tx frame control information is set to 1, the TCPIP accelerator function is switched off for each packet.

Hardware does not calculate the TCP/UDP checksum of fragmented packets. The checksum should be calculated by software.

When the TTCPIPEN bit of the GMAC_ACC register is set to 0, the TCPIP accelerator for transmission is disabled.

Table 3.67 GMAC_ACC Register Settings and Operation of the Tx TCPIP Accelerator

GMAC_ACC.TTCPIPEN	Tx Frame Control Information TCPIP ACC OFF	Checksum Calculation (Tx)	Padding for TCPIPACC (Tx)
0	0	Not available	Not required
0	1	Not available	Not required
1	0	Available	Required
1	1	Not available	Required

NOTE

If the UDP checksum for transmission packets calculated by hardware is 0x0000, the checksum field of the UDP header is changed to 0xFFFF in this packet.

CAUTION

If the value of length field within IPv4 header is not matched to actual frame length, transmission may not be completed successfully. Must be proper value in IPv4 header.

(2) Reception Using the TCPIP Accelerator

If the RTCPIPEN bit of the GMAC_ACC register is set to 1, the Rx TCPIP accelerator function for reception is enabled. If a packet including IPv4, TCP/IP, or UDP/IP is received while the Rx TCPIP accelerator is enabled, hardware automatically calculates the checksum of the packet. If the result of calculation is not equal to the value of the checksum field in the packet, error information is stored in the IPNG or TCPNG field of Rx frame information.

While Rx TCPIPACC is enabled, 2-byte padding for TCPIPACC is inserted in the MAC header of the received frame.

If the Rx TCPIP accelerator function is enabled and the received packet has TCP/UDP, the FCS field is overwritten by the TCP/UDP checksum. This checksum can be used to calculate the total checksum of fragmented TCP/UDP packets.

If any field of IPNG, IPV6NG, or OUT_OF_LIST of Rx frame information shows 1, hardware does not calculate the checksum for the received frame at that time. And also, if the IPv6 extension header includes the fragment, ESP, or AH protocol, TCP/UDP checksum calculation does not proceed.

If the RTCPIPACC bit of the GMAC_ACC register is set to 1, checksum calculation does not proceed but padding for TCPIPACC is inserted in the received frame.

If the RTCPIPEN bit of the GMAC_ACC register is set to 0, the Rx the TCPIP accelerator function for reception is disabled. If this is the case, padding for TCPIPACC is not inserted in the received frame.

Table 3.68 GMAC_ACC Register Settings and Operation of the Rx TCPIP Accelerator

GMAC_ACC. RTCPIPEN	GMAC_ACC. RTCPIPACC	Checksum Calculation (Rx)	Padding for TCPIPACC (Rx)	Checksum Calculated by Hardware Overwrites the FCS Field
0	0	No	No	No
0	1	No	No	No
1	0	Yes	Yes	Yes
1	1	No	Yes	No

NOTE

If the UDP checksum field in the received packet is 0x0000, hardware does not check checksum validation.

3.6 Usage Notes

3.6.1 Giga Bit Ethernet MAC (HW-RTOS GMAC) Control Register

3.6.1.1 MIIM Register (GMAC_MIIM)

The GMAC_MIIM register is used to control register access to each Ethernet PHY. For access to this register, then follow the procedure below.

For write access:

1. Start of a write operation:
   Set the RWDV bit to 1; set the PHYADDR[4:0] bits to the PHY address; set the REGADDR[4:0] bits to the PHY register address; set the DATA[15:0] bits to the write data.
2. Wait for the operation to finish:
   Wait until 1 is read from the RWDV bit.
3. End of the operation:
   The write operation finishes when 1 is read from the RWDV bit.

For read access:

1. Start of a read operation:
   Set the RWDV bit to 0; set the PHYADDR[4:0] bits to the PHY address; set the REGADDR[4:0] bits to the PHY register address.
2. Wait for the operation to finish:
   Wait until 1 is read from the RWDV bit.
3. End of the operation:
   When 1 is read from the RWDV bit and valid data is read from the DATA[15:0] bits, the read operation finishes.

3.6.1.2 TX ID Register (GMAC_TXID)

The GMAC_TXID register is used to indicate the ID of the transmission frame for the GMAC_TXRESULT register. This ID is specified by Frame ID field of Transmit Frame Control Information.

To check the transmission frame result ID, be sure to read this register before reading the GMAC_TXRESULT register. If the GMAC_TXRESULT register is read first, the transmission frame result is updated. Therefore, the updated transmission frame ID is read from the GMAC_TXID register.

When RMTAGCTRL.MGMT_ENB = 1, GMAC_TXID is valid only for bit7 to bit0.

3.6.1.3 TX Result Register (GMAC_TXRESULT)

The GMAC_TXRESULT register is used to indicate the transmission frame result.

The transmission frame result is updated when this register is read. The updated transmission frame result can be read at the next time.

This register indicates the transmission frame result. It is only available while GMAC_TXMODE.TRBMODE[1:0] bits are 00b or 01b.

The transmission frame result is stored in the transmission result buffer when the Ethernet transmission complete interrupt (HWRTOS_ETHIT_Int) occurs. The transmission result buffer can hold 4 frames of information. Reading this register leads to the frame information being removed from the transmission result buffer. The number of frames stored in this buffer can be obtained from the GMAC_TXFIFO.TRBFR bit.

If transmission starts while the transmission result buffer has 4 frames, transmission is invalid and the TX-FIFO error interrupt (HWRTOS_ETHTFIE_Int) occurs. While this register is available, must read it appropriately so that no error occurs.

3.6.1.4 TX Mode Register (GMAC_TXMODE)

The GMAC_TXMODE register is used to control frame transmission.

LPTXEN must be set to 1 if the frame size exceeds the maximum size of 1518 bytes. For example, in case that management tag insertion of the Ethernet switch is enabled (the MGMT_ENB bit in the RMTAGCTRL register is 1).

3.6.1.5 Reset Register (GMAC_RESET)

The GMAC_RESET register is a trigger register that is used to reset the HW-RTOS GMAC module by software. The module is reset by setting ALLRST bits to 1.

The waiting time for the completion of a reset depends on the operating mode of the MAC as listed below.

Operation at 1 Gbps (125 MHz): 60 ns

3.6.1.6 Pause Packet Data Register [m] (GMAC_PAUSE[m])

The register is used to specify the pause packet to be sent.

The transmission packet format is shown below.

	31 – 16	15 – 0
GMAC_PAUSE1	Destination Address	
GMAC_PAUSE2	Source Address	Destination Address
GMAC_PAUSE3	Source Address	
GMAC_PAUSE4	Opcode	Type/Length
GMAC_PAUSE5	(Not used)	Time

Figure 3.16 Transmission packet format using GMAC_PAUSE[m] register

3.6.2 Hardware Function Call Register

The hardware function call register is used to obtain a buffer. This register is also used to start transmission or reception. After argument registers (R4 to R7) are configured, a hardware function is executed by writing a command to the system call register (SYSC). For details on how to configure the hardware function call register, see section Hardware Function.

NOTE

The hardware function related registers are also used for controlling the hardware real-time OS

3.6.3 Management TAG Control

If the management tag function in MGMT_TAG_CONFIG register of A5PSW is enabled between HWRTOS GMAC and the Advanced 5ports SWITCH (A5PSW) management port, RMTAGCTRL setting in Ethernet Accessory Register decides whether to use the management tag in Ethernet frame data (frame pattern B) or to modify the configuration formats of the following information fields (frame pattern C).

- Transmit: Transmit Frame Control Information field
- Receive: Destination MAC Address field

Table 3.69 Frame Pattern with MGMT_TAG_CONFIG and RMTAGCTRL Register Settings

MGMT_TAG_CONFIG	RMTAGCTRL	Frame Pattern	Note
0	0	A	Tag function disabled, Without Tag data insertion
0	1	—	(unused)
1	0	B	Tag function enabled, With Tag data insertion
1	1	C	Tag function enabled, Without Tag data insertion

Figure 3.17 Management TAG configuration

3.6.4 Notice

3.6.4.1 Padding Added to the MAC Header in the Transmit Frame

In the gigabit Ethernet MAC, a transmission frame is normally composed of the 14-byte MAC header plus 2 bytes of padding so that the TCPIP accelerator handles the data.

However, the padding is not actually sent. Accordingly, note that it is not included in the data size of the frame for transmission.

Refer to **Section 3.5.6(1), Transmission Using the TCPIP Accelerator**, for detail.

3.6.4.2 Erroneous Judgment about Checksum Validation at Specific Frame Reception

(1) Frame type: Ethernet II or IEEE 802.3 + IEEE 802.2 (LLC+SNAP)

If frame with any of the following conditions is received, the IPNG or TCPNG field of Rx Frame Information may be set to 1 despite the received packet is valid. In that case, check the checksum by software.

- IPv4 and checksum field value in the TCP header is 0x0000 or 0xFFFF.
- Ethernet frame length except FCS is 60 bytes and over, and the payload size of TCP or UDP on IPv6 is 1 byte, and the data following payload (basically FCS) is not 0.
- The sum value of pseudo header for TCP or UDP on IPv6 is 21 bits and over.

(2) Frame type: IEEE 802.3 + IEEE 802.2 (LLC)

If IEEE802.3 + IEEE802.2 (LLC) frame without SNAP is received, the TYPEIP and IPNG field of Rx Frame Information may be set to 1 despite IP packet is not included. In that case, check by software if SNAP is included or not. If SNAP is not included, consider the received frame valid.

3.6.4.3 Error of Receive Frame Information at Rx FIFO Overflow

If Rx TCPIP accelerator is enabled and Rx FIFO is overflowed, Rx Frame Information might include error as below.

- Error information which is related to the previous error frame is included in frame information of normal reception frame.
- Receive Frame Information may store illegal value because overflowed frame is recognized as valid.

Apply any of the following methods to avoid the issue.

(A) Disable Rx TCPIP accelerator function which also includes padding insertion to MAC header. Specifically, clear bit0 of GMAC_ACC register.

(B) When Rx FIFO is overflowed, discard all received frames left in Rx FIFO and Buffer RAM. Specifically, apply the following procedure;

(1) Disable Rx MAC.

(2) Discard all frames inside Rx FIFO.

(3) Discard all frames inside Buffer RAM.

(4) Enable Rx MAC.

(5) Discard at least one frame with VALID bit = 1 of BUFID register. This is because FIFO empty state can be read even if the frame which caused FIFO overflow remains in the FIFO. Receive normal frame once and discard remained abnormal frame with it.

- In case the hardware real-time OS is used.

  Figure 3.18: Flowchart of RX FIFO Overflow Processing Task

  Figure 3.19: Flowchart of Reception processing

- In case the hardware real-time OS is not used.

  Figure 3.20: Flowchart of RX FIFO overflow interrupt processing

  Figure 3.21: Flowchart of Reception processing

CAUTION

- The abnormal frame remained in the FIFO is discarded in reception processing. Discard valid data once when overflow return flag is set.
- Overflow return flag is a global variable.
- Overflow interrupt is disabled from reading BUFID till checking overflow return flag.

Figure 3.18 Flowchart of RX FIFO Overflow Processing Task (In Case the Hardware Real-Time OS is Used)

Figure 3.19 Flowchart of Reception Processing Task (In Case the Hardware Real-Time OS is Used)

Figure 3.20 Flowchart of RX FIFO Overflow Interrupt Processing (In Case the Hardware Real-Time OS is not Used)

Figure 3.21 Flowchart of Reception Processing (In Case the Hardware Real-Time OS is not Used)

3.6.4.4 Error of Rx Frame Information at Reception of the Frame More than 64 bytes with Padding

If Rx TCPIP accelerator is enabled and the frame meets all the following condition, it is possible that reception word size (RX_WORD[12:0]) in the frame information increases by 1 word (4 bytes) or decreases by 1 word compared with correct size. In case of decrease by 1 word, it is possible that RX_WORD indicates the size which causes lack of IP packet. IP packet itself is NOT lacked.

- Frame size including FCS is more than 64 bytes.
- TCP/IP or UDP/IP packet is included.
- Padding (Trailer) is included between IP packet and FCS.

Apply any of the following methods to avoid the issue

(A) Disable Rx TCPIP accelerator. Specifically, clear bit0 or set bit2 of GMAC_ACC register.

(B) To avoid lack of the IP packet, increase reception word size by 1 and transfer the size to protocol stack. In the protocol stack, payload data should be extracted based on size of Total Length field in IP header and the rest data should be discarded. **Figure 3.22** is the flowchart of this workaround.

Figure 3.22 Flowchart of Reception Processing

3.6.4.5 About Jumbo Frame

HW-RTOS GMAC doesn't support jumbo frame.

Section 4 Advanced 5port Switch (A5PSW)

4.1 Overview

- Switch 5 Ports
  - 1 Port connected to CPU with multiplexor, as a management port (port 4), and enabled at 1000 Mbps full-duplex mode only
    1) Connected through Gigabit Ethernet MAC2 (GMAC2) 1000/100 with Timer IEEE 1588-2008
    2) Or HW-RTOS GMAC
  - The frequency of switch system clock (A5PSW_SXCLK) impacts port speed:
    1) 200 MHz is recommended to use all ports in 1000 Mbps
    2) This means that 4 line-ports and 1 management port are available in 1000 Mbps
       166.6 MHz is enough for 3 ports in 1000 Mbps and 2 ports in 100 Mbps
       This means that 2 line ports in 1000 Mbps and 2 line ports in 100 Mbps and 1 management port in 1000 Mbps are available.
- Supported Communication modes
  - 10 Mbps Half and Full-duplex
  - 100 Mbps Half and Full-duplex
  - 1000 Mbps Full-duplex only
- MAC based RMON (Remote network MONitoring) Statistics counters/per port
  - Half-Duplex counters
    FCS (Frame Check Sequence) Error, Deferred, Single, Multiple, Excessive, Late
- Port statistics on per port basis (no aggregation)
- Implements hardware switching look up mechanism providing a learning capacity of 8192 MAC addresses (including static and learned)
  - Programmable firmware operation with Static or Dynamic (Learning, Aging) switching tables
- Packet buffer size: 1 Mbit (important for leaky bucket, QoS, etc.)
- 4 queues with individual QoS levels, supporting frame priority classification for the flexible handling of output queues
  - Classification and Priority assignment based on Port Number, MAC Address, IPv4 DiffServ Code Point (DSCP) Field, IPv6 Traffic Class and VLAN Priority (IEEE 802.1Q).
  - Optional arbitration management through weighted fair queuing with fixed weights of 1-2-4-8.
- Support for Ethernet multicast and broadcast frames with flooding control to avoid unnecessary duplication of frames
- Programmable multicast destination port mask to restrict frame duplication for individual multicast addresses
- IEEE 1588-2008 support providing for frame timestamp forwarding enabling IEEE 1588 applications implementing ordinary, boundary or transparent clocks
  - Support for 1 step Peer to Peer (P2P) (Layer 2 only)
  - Support for 1 step End to End (E2E) (Layer 2 only)
- Multicast and broadcast resolution with VLAN domain filtering providing a strict separation of up to 32 VLANs
- Support for reception and transmission of VLAN frames

- Programmable addition, removal and manipulation of ingress and egress VLAN tags, supporting single and double-tagged VLAN frames on each port
- Whenever the DA (Destination Address) is unknown, it will always flood only to those ports that are listed in the VLAN
- Support for standard frame size (1536 bytes), extended frame sizes up to 1700 bytes and jumbo frames up to 10 Kbytes
- Switch forwarding delay if switch system clock is 200 MHz:
  - From packet end to packet start 64 bytes → 1.6 µs
  - From packet end to packet start 1518 bytes → 2.8 µs

NOTE

All times measured at the switch's MIIs. PHY latencies as well as interface conversion latencies (e.g. RMII, RGMII) will add to these numbers.

- Propagation time/port latency if switch system clock is 200 MHz (8-byte preamble considered):
  - From packet start to packet start 64 bytes for 100 Mbps → 72 × 80 ns + 1.6 µs → to 7.4 µs
  - From packet start to packet start 1518 bytes for 100 Mbps → 1526 × 80 ns + 2.8 µs → to 125 µs

NOTE

All times measured at the switch's MIIs. PHY latencies as well as interface conversion latencies (e.g. RMII, RGMII) will add to these numbers.

- Port mirroring programmable per port
- RSTP port states (3 for RSTP/ 5 for STP)
  - RSTP Port states learning, discarding, forwarding configurable per port
  - BPDU frame supported
  - MSTP BPDU frame supported (software)
- Configured to start the switch in managed mode (all ports are OFF until the CPU configures)
- Frame snooping engine
  - Allows IGMP snooping to the software.
- Stand-alone Energy Efficient Ethernet (EEE) management
  - Enable & configure register per port
  - 2 hardware timers per ports
  - Two possible modes:
    1) Fully automatic operation
    2) Software controlled operation
- Egress rate limit per port
  - Programmable, 10% to 80% bandwidth.

- Ingress Configurable Broadcast storm protection per port
  - To avoid network and memory congestion caused by a broadcast storm on the MAC Interfaces, every port can be programmed to limit the number of broadcast frames over a certain time period.
  - A counter is implemented on every MAC interface receive port, which increments whenever a broadcast frame is received. The counter is cleared each time the timeout expires.
  - When the counter reaches the limit, further broadcast frames are discarded.
- Ingress Configurable Multicast storm protection per port
  - To avoid network and memory congestion caused by a multicast storm on the MAC Interfaces, every port can be programmed to limit the number of multicast frames over a certain time period.
  - A counter is implemented on every MAC interface receive port, which increments whenever a multicast frame is received. The counter is cleared each time the timeout expires.
  - When the counter reaches the limit, further multicast frames are discarded.
- 802.1X source address authentication supported
- 802.1X guest VLAN supported
- PRP functionality (IEC 62439-3 edition 2.0 - 2012)
- DLR module (Port A and B)
- Cut through
  - Jumbo frame feature available between two ports in cut through
  - The cut-through forwarding delay depends on link speed and the switch system clock frequency.
    If the switch system clock frequency is 200 MHz:

Link Speed	Forwarding Delay (SFD to SFD, without PHY Latencies)
10 Mbps	Up to 35 µs
100 Mbps	3.2 … 3.8 µs (no pattern matcher active) 3.5 … 3.9 µs (pattern matcher active)
1 Gbps	550 … 800 ns (pattern matcher active)

  - Pattern Matcher will delay the cut-through decision slightly (by up to 4 bytes) preventing the underflow and thus frame corruption.
- TDMA (Time Division Multiple Access)
  - Allow to support network infrastructures that define a time multiplexed access to reserve bandwidth for different traffic classes, the switch can be configured to operate some or all ports.
- Pattern Matchers 8 channels
- Remote monitoring via SNMP and the (RMON / MIB)
- Interface
  - Native mode GMII/MII
  - No Native RMII, RGMII managed by RMII / RGMII convertor (connected on external pins)
- Hub on ports A, B, C and D
  - Switch HUB function
    Scheduler controlling that only one MAC can transmit to the Hub ports and global collision detection when multiple ports receive while not transmitting (following IEEE 802.3 Clause 27 behavior).

– PHY interface Hub Function
Copies all frames in between the PHY interfaces without going through the switch to allow meeting minimum latency requirements.
Support for RMII only

Figure 4.1 A5PSW Block Diagram

4.2 Signal Interface

Table 4.1 Signal Interface of the A5PSW (excluding PHY MII pins)

Signal Name	I/O	Description	Active
Clock			
A5PSW_HCLK	I	AHB clock used for the register interface.	
A5PSW_SXCLK	I	Switch system clock.	
Signals for PTP			
A5PSW_TSCLK	I	Timer Module clock. All timestamp related functions operate on this clock. The clock source is selected by PTP Mode Control Register.	
A5PSW_TS_NS_IN[31:0]	I	Current time used for all timestamping functions. (nanoseconds)	
Interrupt			
A5PSW_Int	O	A5PSW Interrupt, level sensitive	High
A5PSW_PRP_Int	O	A5PSW – PRP interrupt, level sensitive	High
A5PSW_HUB_Int	O	A5PSW – Integrated Hub module, level sensitive	High
A5PSW_PTRN_Int	O	A5PSW – RX Pattern Matcher, level sensitive	High
A5PSW_DLR_Int	O	A5PSW – DLR interrupt, level sensitive	High
External Signal			
SWITCH_MII_LINK[5]	I	Link status from PHY. (port A) (Port 0 of A5PSW)	High*1
SWITCH_MII_LINK[4]	I	Link status from PHY. (port B) (Port 1 of A5PSW)	High*1
SWITCH_MII_LINK[3]	I	Link status from PHY. (port C) (Port 2 of A5PSW)	High*1
SWITCH_MII_LINK[2]	I	Link status from PHY. (port D) (Port 3 of A5PSW)	High*1

Note 1. SWITCH_MII_LINK[5:2] active level is controlled by Ethernet PHY Link Mode register.

4.3 Register Map

Table 4.2 A5PSW Register Map (1/6)

Address	Register Symbol	Register Name
4405 0000h	REVISION	Switch Core Version
4405 0004h	SCRATCH	Scratch Register
4405 0008h	PORT_ENA	Port Enable Register
4405 000Ch	UCAST_DEFAULT_MASK	Unicast Default Mask Register
4405 0010h	VLAN_VERIFY	Verify VLAN domain
4405 0014h	BCAST_DEFAULT_MASK	Broadcast Default Mask Register
4405 0018h	MCAST_DEFAULT_MASK	Multicast Default Mask Register
4405 001Ch	INPUT_LEARN_BLOCK	Input Learning Block Register
4405 0020h	MGMT_CONFIG	Management Configuration Register
4405 0024h	MODE_CONFIG	Mode Configuration Register
4405 0028h	VLAN_IN_MODE	VLAN input manipulation mode Register
4405 002Ch	VLAN_OUT_MODE	VLAN output manipulation mode Register
4405 0030h	VLAN_IN_MODE_ENA	VLAN input mode enable Register
4405 0034h	VLAN_TAG_ID	VLAN Tag ID Register
4405 0038h	BCAST_STORM_LIMIT	Broadcast Storm Protection Register
4405 003Ch	MCAST_STORM_LIMIT	Multicast Storm Protection Register
4405 0040h	MIRROR_CONTROL	Port Mirroring Configuration Register
4405 0044h	MIRROR_EG_MAP	Port Mirroring Egress Port Definition
4405 0048h	MIRROR_ING_MAP	Port Mirroring Ingress Port Definition
4405 004Ch	MIRROR_ISRC_0	Ingress Source MAC Address for Mirror Filtering 0
4405 0050h	MIRROR_ISRC_1	Ingress Source MAC Address for Mirror Filtering 1
4405 0054h	MIRROR_IDST_0	Ingress Destination MAC Address for Mirror Filtering 0
4405 0058h	MIRROR_IDST_1	Ingress Destination MAC Address for Mirror Filtering 1
4405 005Ch	MIRROR_ESRC_0	Egress Source MAC Address for Mirror Filtering 0
4405 0060h	MIRROR_ESRC_1	Egress Source MAC Address for Mirror Filtering 1
4405 0064h	MIRROR_EDST_0	Egress Destination MAC Address for Mirror Filtering 0
4405 0068h	MIRROR_EDST_1	Egress Destination MAC Address for Mirror Filtering 1
4405 006Ch	MIRROR_CNT	Mirror Filtering Count Value Register
4405 0088h	QMGR_ST_MINCELLS	Output Queue Minimum Memory Statistics Register
4405 0094h	QMGR_RED_MIN4	RED Minimum Threshold Register
4405 0098h	QMGR_RED_MAX4	RED Maximum Threshold Register
4405 009Ch	QMGR_RED_CONFIG	RED Configuration Register
4405 00A0h	IMC_STATUS	Input Memory Controller Status Register
4405 00A4h	IMC_ERR_FULL	Input Port Memory Full and Truncation Indicator
4405 00A8h	IMC_ERR_IFACE	Input Port Memory Error Indicator
4405 00ACh	IMC_ERR_QOFLOW	Output Port Queue Overflow Indicator
4405 00B0h	IMC_CONFIG	Input Memory Controller Configuration Register
4405 00C0h+4h×n	GPARSER[n] (n=0..3)	[n]th parser of 1st block
4405 00D0h+4h×n	GARITH[n] (n=0..3)	Snoop Configuration for Arithmetic [n]th Stage of 1st block
4405 00E0h+4h×(n−4)	GPARSER[n] (n=4..7)	[n−4]th parser of 2nd block
4405 00F0h+4h×(n−4)	GARITH[n] (n=4..7)	Snoop Configuration for Arithmetic [n−4]th Stage of 2nd block
4405 0100h+4h×n	VLAN_PRIORITY[n] (n=0..4)	VLAN Priority Register [n]
4405 0140h+4h×n	IP_PRIORITY[n] (n=0..4)	IP Priority Register [n]

Table 4.2 A5PSW Register Map (2/6)

Address	Register Symbol	Register Name
4405 0180h +4h*n	PRIORITY_CFG[n] (n=0..4)	Priority Configuration Register [n]
4405 01B8h	PRIORITY_TYPE1	Priority Type Register 1
4405 01BCh	PRIORITY_TYPE2	Priority Type Register 2
4405 01C0h	MGMT_ADDR0_lo	Lower MAC Address for Bridge Protocol Frame
4405 01C4h	MGMT_ADDR0_hi	Higher MAC Address for Bridge Protocol Frame
4405 0200h+4h×n	SYSTEM_TAGINFO[n] (n=0..4)	One VLAN ID field for VLAN input manipulation
4405 0240h+4h×n	AUTH_PORT[n] (n=0..4)	PORT [n] authentication control and configuration
4405 0280h+4h×n	VLAN_RES_TABLE[n] (n=0..31)	32 VLAN domain entries
4405 0300h	TOTAL_DISC	Discarded Frame Total Number Register
4405 0304h	TOTAL_BYT_DISC	Discarded Frame Total Bytes Register
4405 0308h	TOTAL_FRM	Processed Frame Total Number Register
4405 030Ch	TOTAL_BYT_FRM	Processed Frame Total Bytes Register
4405 0310h+10h×n	ODISC[n] (n=0..4)	PORT[n] Discarded Outgoing Frame Count Register
4405 0314h+10h×n	IDISC_VLAN[n] (n=0..4)	PORT[n] Discarded Incoming VLAN Tagged Frame Count Register
4405 0318h+10h×n	IDISC_UNTAGGED[n] (n=0..4)	PORT[n] Discarded Incoming VLAN Untagged Frame Count Register
4405 031Ch+10h×n	IDISC_BLOCKED[n] (n=0..4)	PORT[n] Discarded Incoming Blocked Frame Count Register
4405 03C0h+4h×n	IMC_QLEVEL_P[n] (n=0..4)	PORT[n] Queued Frame Count Register
4405 0400h	LK_CTRL	Learning/Lookup Function Global Configuration Register
4405 0404h	LK_STATUS	Status bits and table overflow counter
4405 0408h	LK_ADDR_CTRL	Address table transaction control and read/write address
4405 040Ch	LK_DATA_LO	Lower 32-bit data of Lookup Memory Entry
4405 0410h	LK_DATA_HI	Higher 26-bit data of Lookup Memory Entry
4405 0414h	(Reserved)	
4405 0418h	LK_LEARNCOUNT	Learned Address Count Register
4405 041Ch	LK_AGETIME	Period of the aging timer
4405 0480h	MGMT_TAG_CONFIG	Management Tag Configuration Register
4405 050Ch+4h×n	PEERDELAY[n] (n=0..3)	Peer delay value for PORT[n]
4405 0520h+8h×n	PORT[n]_CTRL (n=0..3)	PORT[n] timestamp control/status
4405 0524h+8h×n	PORT[n]_TIME (n=0..3)	PORT[n] Memorized transmit timestamp
4405 0600h	INT_CONFIG	Interrupt Enable Configuration Register
4405 0604h	INT_STAT_ACK	Interrupt Status/ACK Register
4405 0700h	MDIO_CFG_STATUS	MDIO Configuration and Status Register
4405 0704h	MDIO_COMMAND	MDIO PHY Command Register
4405 0708h	MDIO_DATA	MDIO Data Register
4405 0800h+400h×n	REV_P[n] (n=0..4)	PORT[n] MAC Core revision
4405 0808h+400h×n	COMMAND_CONFIG_P[n] (n=0..4)	PORT[n] Command Configuration Register
4405 080Ch+400h×n	MAC_ADDR_0_P[n] (n=0..3)	PORT[n] MAC Address Register 0
4405 0810h+400h×n	MAC_ADDR_1_P[n] (n=0..3)	PORT[n] MAC Address Register 1
4405 0814h+400h×n	FRM_LENGTH_P[n] (n=0..4)	PORT[n] Maximum Frame Length Register
4405 0818h+400h×n	PAUSE_QUANT_P[n] (n=0..4)	PORT[n] MAC Pause Quanta
4405 0830h+400h×n	PTPClockIdentity1_P[n] (n=0..3)	PORT[n] PTP Clock Identity1 Register
4405 0834h+400h×n	PTPClockIdentity2_P[n] (n=0..3)	PORT[n] PTP Clock Identity2 Register
4405 0838h+400h×n	PTPAutoResponse_P[n] (n=0..3)	PORT[n] PTP Auto Response Register

Table 4.2 A5PSW Register Map (3/6)

Address	Register Symbol	Register Name
4405 0840h+400h×n	STATUS_P[n] (n=0..4)	PORT[n] Port Status Register
4405 0844h+400h×n	TX_IPG_LENGTH_P[n] (n=0..4)	PORT[n] Transmit IPG Length Register
4405 0848h+400h×n	EEE_CTL_STAT_P[n] (n=0..3)	PORT[n] MAC EEE functions control and status
4405 084Ch+400h×n	EEE_IDLE_TIME_P[n] (n=0..3)	PORT[n] EEE Idle Time Register
4405 0850h+400h×n	EEE_TWSYS_TIME_P[n] (n=0..3)	PORT[n] EEE Wake Up Time Register
4405 0854h+400h×n	IDLE_SLOPE_P[n] (n=0..4)	PORT[n] MAC Traffic Shaper bandwidth control
4405 0868h+400h×n	aFramesTransmittedOK_P[n] (n=0..4)	PORT[n] MAC Transmitted Valid Frame Count Register
4405 086Ch+400h×n	aFramesReceivedOK_P[n] (n=0..4)	PORT[n] MAC Received Valid Frame Count Register
4405 0870h+400h×n	aFrameCheckSequenceErrors_P[n] (n=0..4)	PORT[n] MAC FCS Error Frame Count Register
4405 0874h+400h×n	aAlignmentErrors_P[n] (n=0..4)	PORT[n] MAC Alignment Error Frame Count Register
4405 0878h+400h×n	aOctetsTransmittedOK_P[n] (n=0..4)	PORT[n] MAC Transmitted Valid Frame Octets Register
4405 087Ch+400h×n	aOctetsReceivedOK_P[n] (n=0..4)	PORT[n] MAC Received Valid Frame Octets Register
4405 0880h+400h×n	aTxPAUSEMACCtrlFrames_P[n] (n=0..4)	PORT[n] MAC Transmitted Pause Frame Count Register
4405 0884h+400h×n	aRxPAUSEMACCtrlFrames_P[n] (n=0..4)	PORT[n] MAC Received Pause Frame Count Register
4405 0888h+400h×n	ifInErrors_P[n] (n=0..4)	PORT[n] MAC Input Error Count Register
4405 088Ch+400h×n	ifOutErrors_P[n] (n=0..4)	PORT[n] MAC Output Error Count Register
4405 0890h+400h×n	ifInUcastPkts_P[n] (n=0..4)	PORT[n] MAC Received Unicast Frame Count Register
4405 0894h+400h×n	ifInMulticastPkts_P[n] (n=0..4)	PORT[n] MAC Received Multicast Frame Count Register
4405 0898h+400h×n	ifInBroadcastPkts_P[n] (n=0..4)	PORT[n] MAC Received Broadcast Frame Count Register
4405 089Ch+400h×n	ifOutDiscards_P[n] (n=0..4)	PORT[n] MAC Discarded Outbound Frame Count Register (Not Applicable)
4405 08A0h+400h×n	ifOutUcastPkts_P[n] (n=0..4)	PORT[n] MAC Transmitted Unicast Frame Count Register
4405 08A4h+400h×n	ifOutMulticastPkts_P[n] (n=0..4)	PORT[n] MAC Transmitted Multicast Frame Count Register
4405 08A8h+400h×n	ifOutBroadcastPkts_P[n] (n=0..4)	PORT[n] MAC Transmitted Broadcast Frame Count Register
4405 08Ach+400h×n	etherStatsDropEvents_P[n] (n=0..4)	PORT[n] MAC Dropped Frame Count Register
4405 08B0h+400h×n	etherStatsOctets_P[n] (n=0..4)	PORT[n] MAC All Frame Octets Register
4405 08B4h+400h×n	etherStatsPkts_P[n] (n=0..4)	PORT[n] MAC All Frame Count Register
4405 08B8h+400h×n	etherStatsUndersizePkts_P[n] (n=0..4)	PORT[n] MAC Too Short Frame Count Register
4405 08BCh+400h×n	etherStatsOversizePkts_P[n] (n=0..4)	PORT[n] MAC Too Long Frame Count Register
4405 08C0h+400h×n	etherStatsPkts64Octets_P[n] (n=0..4)	PORT[n] MAC 64 Octets Frame Count Register
4405 08C4h+400h×n	etherStatsPkts65to127Octets_P[n] (n=0..4)	PORT[n] MAC 65 to 127 Octets Frame Count Register
4405 08C8h+400h×n	etherStatsPkts128to255Octets_P[n] (n=0..4)	PORT[n] MAC 128 to 255 Octets Frame Count Register
4405 08CCh+400h×n	etherStatsPkts256to511Octets_P[n] (n=0..4)	PORT[n] MAC 256 to 511 Octets Frame Count Register
4405 08D0h+400h×n	etherStatsPkts512to1023Octets_P[n] (n=0..4)	PORT[n] MAC 512 to 1023 Octets Frame Count Register
4405 08D4h+400h×n	etherStatsPkts1024to1518Octets_P[n] (n=0..4)	PORT[n] MAC 1024 to 1519 Octets Frame Count Register
4405 08D8h+400h×n	etherStatsPkts1519toXOctets_P[n] (n=0..4)	PORT[n] MAC Over 1519 Octets Frame Count Register
4405 08DCh+400h×n	etherStatsJabbers_P[n] (n=0..4)	PORT[n] MAC Jabbers Frame Count Register
4405 08E0h+400h×n	etherStatsFragments_P[n] (n=0..4)	PORT[n] MAC Fragment Frame Count Register
4405 08E8h+400h×n	VLANReceivedOK_P[n] (n=0..4)	PORT[n] MAC Received VLAN Tagged Frame Count Register
4405 08F4h+400h×n	VLANTransmittedOK_P[n] (n=0..4)	PORT[n] MAC Transmitted VLAN Tagged Frame Count Register
4405 08F8h+400h×n	FramesRetransmitted_P[n] (n=0..4)	PORT[n] MAC Retransmitted Frame Count Register
4405 0900h+400h×n	STATS_HIWORD_P[n] (n=0..4)	PORT[n] MAC Statistics Counter High Word Register
4405 0904h+400h×n	STATS_CTRL_P[n] (n=0..4)	PORT[n] MAC Statistics Control Register

Table 4.2 A5PSW Register Map (4/6)

Address	Register Symbol	Register Name
4405 0908h+400h×n	STATS_CLEAR_VALUELO_P[n] (n=0..4)	PORT[n] MAC Statistics Clear Value Lower Register
4405 090Ch+400h×n	STATS_CLEAR_VALUEHI_P[n] (n=0..4)	PORT[n] MAC Statistics Clear Value Higher Register
4405 0910h+400h×n	aDeferred_P[n] (n=0..3)	PORT[n] MAC Deferred Count Register
4405 0914h+400h×n	aMultipleCollisions_P[n] (n=0..3)	PORT[n] MAC Multiple Collision Count Register
4405 0918h+400h×n	aSingleCollisions_P[n] (n=0..3)	PORT[n] MAC Single Collision Count Register
4405 091Ch+400h×n	aLateCollisions_P[n] (n=0..3)	PORT[n] MAC Late Collision Count Register
4405 0920h+400h×n	aExcessiveCollisions_P[n] (n=0..3)	PORT[n] MAC Excessive Collision Count Register
4405 0924h+400h×n	aCarrierSenseErrors_P[n] (n=0..3)	PORT[n] MAC Carrier Sense Error Count Register
4405 3C00h	DLR_CONTROL	DLR Control Register
4405 3C04h	DLR_STATUS	DLR Status Register
4405 3C08h	DLR_ETH_TYP	DLR Ethernet Type Register
4405 3C0Ch	DLR_IRQ_CONTROL	DLR Interrupt Control Register
4405 3C10h	DLR_IRQ_STAT_ACK	DLR Interrupt Status/ACK Register
4405 3C14h	DLR_LOC_MAClo	DLR Local MAC Address Low Register
4405 3C18h	DLR_LOC_MAChi	DLR Local MAC Address High Register
4405 3C20h	DLR_SUPR_MAClo	DLR Supervisor MAC Address Low Register
4405 3C24h	DLR_SUPR_MAChi	DLR Supervisor MAC Address High Register
4405 3C28h	DLR_STATE_VLAN	DLR Ring Status/VLAN Register
4405 3C2Ch	DLR_BEC_TMOUT	DLR Beacon Timeout Register
4405 3C30h	DLR_BEC_INTRVL	DLR Beacon Interval Register
4405 3C34h	DLR_SUPR_IPADR	DLR Supervisor IP Address Register
4405 3C38h	DLR_ETH_STYP_VER	DLR Sub Type/Protocol Version Register
4405 3C3Ch	DLR_INV_TMOUT	DLR Beacon Timeout Timer Register
4405 3C40h	DLR_SEQ_ID	DLR Sequence ID Register
4405 3C58h	DLR_DSTlo	DLR Beacon Destination Address Low Register
4405 3C5Ch	DLR_DSThi	DLR Beacon Destination Address High Register
4405 3C60h	DLR_RX_STAT0	DLR Received Frame Statistic Register 0
4405 3C64h	DLR_RX_ERR_STAT0	DLR Received Frame Error Statistic Register 0
4405 3C68h	DLR_TX_STAT0	DLR Transmitted Frame Statistic Register 0
4405 3C70h	DLR_RX_STAT1	DLR Received Frame Statistic Register 1
4405 3C74h	DLR_RX_ERR_STAT1	DLR Received Frame Error Statistic Register 1
4405 3C78h	DLR_TX_STAT1	DLR Transmitted Frame Statistic Register 1
4405 3D00h	PRP_CONFIG	PRP Configuration Register
4405 3D04h	PRP_GROUP	PRP Port Group Register
4405 3D08h	PRP_SUFFIX	PRP RCT Suffix
4405 3D0Ch	PRP_LANID	PRP LAN Identifier
4405 3D10h	DUP_W	PRP Max Duplicate Detection Window Size
4405 3D14h	PRP_AGETIME	PRP Aging Time Define Register
4405 3D18h	PRP_IRQ_CONTROL	PRP Interrupt Control Register
4405 3D1Ch	PRP_IRQ_STAT_ACK	PRP Interrupt Status/ACK Register
4405 3D20h	RM_ADDR_CTRL	PRP History Memory Transactions Control Register
4405 3D24h	RM_DATA	PRP Memory Data Register
4405 3D2Ch	RM_STATUS	PRP Memory Controller Status Indication
4405 3D30h	TxSeqTooLate	PRP Frame Transmission Retrieval of Failed Sequence
4405 3D34h	CntErrWrongLanA	PRP Wrong ID LAN-A Count Register

Table 4.2 A5PSW Register Map (5/6)

Address	Register Symbol	Register Name
4405 3D38h	CntErrWrongLanB	PRP Wrong ID LAN-B Count Register
4405 3D3Ch	CntDupLanA	PRP Duplicate LAN-A Count Register
4405 3D40h	CntDupLanB	PRP Duplicate LAN-B Count Register
4405 3D44h	CntOutOfSeqLowA	PRP Sequence Error Low LAN-A Count Register
4405 3D48h	CntOutOfSeqLowB	PRP Sequence Error Low LAN-B Count Register
4405 3D4Ch	CntOutOfSeqA	PRP Sequence Error LAN-A Count Register
4405 3D50h	CntOutOfSeqB	PRP Sequence Error LAN-B Count Register
4405 3D54h	CntAcceptA	PRP Valid Frame LAN-A Count Register
4405 3D58h	CntAcceptB	PRP Valid Frame LAN-B Count Register
4405 3D5Ch	CntMissing	PRP Drop history Adjustment Count
4405 3E00h	HUB_CONFIG	HUB Configuration Register
4405 3E04h	HUB_GROUP	HUB Port Group Register
4405 3E08h	HUB_DEFPORT	HUB Default Port Selection Register
4405 3E0Ch	HUB_TRIGGER_IMMEDIATE	HUB Transmission Trigger Immediate Register
4405 3E10h	HUB_TRIGGER_AT	HUB Transmission Trigger At Register
4405 3E14h	HUB_TTIME	HUB Transmission Time Define Register
4405 3E18h	HUB_IRQ_CONTROL	HUB Interrupt Control Register
4405 3E1Ch	HUB_IRQ_STAT_ACK	HUB Interrupt Status/ACK Register
4405 3E20h	HUB_STATUS	HUB Status Register
4405 3E24h	HUB_OPORT_STATUS	HUB Output Port Status Register
4405 3E80h+4h×n	RXMATCH_CONFIG[n] (n=0..4)	RX Pattern Match Configuration for PORT[n]
4405 3EB0h+4h×n	PATTERN_CTRL[n] (n=0..7)	RX Pattern Match Function Control for Pattern[n]
4405 3ED0h	PTN_IRQ_CONTROL	RX Pattern Match Interrupt Control Register
4405 3ED4h	PTN_IRQ_STAT_ACK	RX Pattern Match Interrupt Status/ACK Register
4405 3EDCh	PATTERN_SEL	RX Pattern Number Selection Register
4405 3EE0h	PTRN_CMP_30	Pattern compare value bytes 3 .. 0
4405 3EE4h	PTRN_CMP_74	Pattern compare value bytes 7 .. 4
4405 3EE8h	PTRN_CMP_118	Pattern compare value bytes 11 .. 8
4405 3EF0h	PTRN_MSK_30	Pattern mask for bytes 3 .. 0.
4405 3EF4h	PTRN_MSK_74	Pattern mask for bytes 7 .. 4.
4405 3EF8h	PTRN_MSK_118	Pattern mask for bytes 11 .. 8.
4405 3F00h	TDMA_CONFIG	TDMA Configuration Register
4405 3F04h	TDMA_PORTS	TDMA Scheduling Enable Register
4405 3F08h	TDMA_START	TDMA Start Time Set Register
4405 3F0Ch	TDMA_MODULO	TDMA System Timer Modulo
4405 3F10h	TDMA_CYCLE	TDMA Periodic Cycle Set Register
4405 3F14h	TDMA_T1	TDMA 1st Time Offset
4405 3F18h	TDMA_T2	TDMA 2nd Time Offset
4405 3F1Ch	TDMA_T3	TDMA 3rd Time Offset
4405 3F20h	QUEUES_TS	TDMA 1st slot Transmit Enable
4405 3F24h	QUEUES_T1	TDMA 2nd slot Transmit Enable
4405 3F28h	QUEUES_T2	TDMA 3rd slot Transmit Enable
4405 3F2Ch	QUEUES_T3	TDMA last slot Transmit Enable
4405 3F30h	QUEUES_START	TDMA First Cycle Transmit Enable
4405 3F34h	TIME_LOAD_NEXT	TDMA Calculated Next Loading Time

Table 4.2 A5PSW Register Map (6/6)

Address	Register Symbol	Register Name
4405 3F38h	TDMA_IRQ_CONTROL	TDMA Interrupt Control Register
4405 3F3Ch	TDMA_IRQ_STAT_ACK	TDMA Interrupt Status/ACK Register

4.4 Register Description

4.4.1 REVISION — Switch Core Version

Address: 4405 0000h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	REV															
Value after reset	0	1	0	1	0	0	0	0	0	0	0	0	0	1	1	1

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	REV															
Value after reset	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0

Table 4.3 REVISION Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	REV	Revision	R

4.4.2 SCRATCH — Scratch Register

Address: 4405 0004h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	SCRATCH															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SCRATCH															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.4 SCRATCH Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	SCRATCH	The Scratch Register provides a memory location to test the register access. It returns all data written to it in inverted form.	R/W

4.4.3 PORT_ENA — Port Enable Register

The enable bits (both per port bits) reset defaults are controlled by STRAP_SX_ENB bit and STRAP_HUB_ENB bit of SWCTRL register.

STRAP_SX_ENB = 0, STRAP_HUB_ENB = 0: RXENA = 0h, TXENA = 0h.
STRAP_SX_ENB = 0, STRAP_HUB_ENB = 1: RXENA = 3h, TXENA = 3h.
STRAP_SX_ENB = 1, STRAP_HUB_ENB = 0: RXENA = fh, TXENA = fh.
STRAP_SX_ENB = 1, STRAP_HUB_ENB = 1: RXENA = 1fh, TXENA = 1fh.
STRAP_SX_ENB and STRAP_HUB_ENB are described in [**Section 10.2.14, SWCTRL — A5PSW Control Register**].

Address: 4405 0008h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	RXENA				
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	TXENA				
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Table 4.5 PORT_ENA Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b21	Reserved	Set to zero on Write. ignore on Read.	R
b20 to b16	RXENA	Receive Enable Mask One bit per port. The switch will serve only input ports that are enabled in this mask	R/W
b15 to b5	Reserved	Set to zero on Write. ignore on Read.	R
b4 to b0	TXENA	Transmit Enable Mask One bit per port. The switch will transmit frames only to ports enabled in this mask. Frames directed to a disabled port are discarded. The exception is all BPDU frames originating from the management port, to which this mask does not apply, allowing to transmit frames to disabled ports (see **Section 4.4.9, MGMT_CONFIG — Management Configuration Register**).	R/W

4.4.4 UCAST_DEFAULT_MASK — Unicast Default Mask Register

Address: 4405 000Ch

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	UCASTDM				
Value after reset	X	X	X	X	X	X	X	X	X	X	X	1	1	1	1	1

Table 4.6 UCAST_DEFAULT_MASK Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b5	Reserved	Set to zero on Write. ignore on Read.	R
b4 to b0	UCASTDM	Default Unicast Resolution 1 bit per port. For unicast frame flooding resolution, the default output port list if the destination address is not found in the MAC lookup table	R/W

4.4.5 VLAN_VERIFY — Verify VLAN Domain

Address: 4405 0010h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	VLANDISC				
Value after reset	X	X	X	X	X	X	X	X	X	X	X	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	VLANVERI				
Value after reset	X	X	X	X	X	X	X	X	X	X	X	0	0	0	0	0

Table 4.7 VLAN_VERIFY Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b21	Reserved	Set to zero on Write. ignore on Read.	R
b20 to b16	VLANDISC	1 bit per port (Bit 16 = Port 0): Discard Unknown. When set, and a frame is received with a VLAN ID that is unknown, or has no VLAN tag, the frame is discarded and not forwarded (i.e. the default broadcast is ignored).	R/W
b15 to b5	Reserved	Set to zero on Write. ignore on Read.	R
b4 to b0	VLANVERI	1 bit per port (Bit 0 = Port 0): Verify VLAN Domain. When enabled (Bit = 1), a frame is accepted from the port as valid only when the input port is a member of the VLAN domain of the frame. When disabled (Bit = 0), frames are forwarded without VLAN domain checking.	R/W

4.4.6 BCAST_DEFAULT_MASK — Broadcast Default Mask Register

Address: 4405 0014h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	BCASTDM				
Value after reset	X	X	X	X	X	X	X	X	X	X	X	1	1	1	1	1

Table 4.8 BCAST_DEFAULT_MASK Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b5	Reserved	Set to zero on Write. ignore on Read.	R
b4 to b0	BCASTDM	Default Broadcast Resolution 1 bit per port. For broadcast/flooding resolution, the default output port list.	R/W

4.4.7 MCAST_DEFAULT_MASK — Multicast Default Mask Register

Address: 4405 0018h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	MCASTDM				
Value after reset	X	X	X	X	X	X	X	X	X	X	X	1	1	1	1	1

Table 4.9 MCAST_DEFAULT_MASK Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b5	Reserved	Set to zero on Write. ignore on Read.	R
b4 to b0	MCASTDM	Default Multicast Resolution 1 bit per port. Used for multicast/flooding resolution. The default output port list used instead of the BCAST_DEFAULT_MASK, when the received frame carries a multicast address.	R/W

4.4.8 INPUT_LEARN_BLOCK — Input Learning Block Register

When blocking is enabled for a port (bit = 1), only Bridge Protocol data units are accepted on that input, all other frames are discarded.

When learning is disabled for a port (bit = 1), source address lookup will not occur and no learning will happen.

Both functions operate independently from each other.

NOTE

Source addresses from incoming BPDU frames are never learned, independent of these settings (to avoid table updates if e.g. loops exist).

Address: 4405 001Ch

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	LEARNDIS				
Value after reset	X	X	X	X	X	X	X	X	X	X	X	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	BLOCKEN				
Value after reset	X	X	X	X	X	X	X	X	X	X	X	0	0	0	0	0

Table 4.10 INPUT_LEARN_BLOCK Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b21	Reserved	Set to zero on Write. ignore on Read.	R
b20 to b16	LEARNDIS	1 bit per port, disable learning (Bit 16 = Port 0)	R/W
b15 to b5	Reserved	Set to zero on Write. ignore on Read.	R
b4 to b0	BLOCKEN	1 bit per port, enable blocking (Bit 0 = Port 0)	R/W

4.4.9 MGMT_CONFIG — Management Configuration Register

Enables and defines the management port that receives Bridge Protocol Frames.

Address: 4405 0020h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	portmask				
Value after reset	X	X	X	X	X	X	X	X	X	X	X	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	priority			—	—	—	—	—	discard	enable	message_transmitted	—	Port			
Value after reset	0	0	0	X	X	X	X	X	1	0	0	X	0	1	0	0

Table 4.11 MGMT_CONFIG Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b21	Reserved	Set to zero on Write. ignore on Read.	R
b20 to b16	portmask	Portmask for transmission of management frames. When the management port transmits a frame to the switch it is forwarded to all ports in this portmask (bit16=port0, bit17=port1, ... bit20=port4). When the mask is all zero the frame is forwarded normally (i.e. either from MAC address lookup or multicast flooding) ***Note)*** Forced forwarding takes precedence over this port mask and management frames will be forwarded as specified by the forced forwarding portmask if active. When using forced forwarding, this portmask must be written with all 0.	R/W
b15 to b13	priority	Priority to use for transmitted BPDU frames if non zero. The setting is ignored if it is 0 (i.e. then the normal priority resolution operates). Can be used to e.g. put a management frame in a high-priority output queue for fast delivery.	R/W
b12 to b8	Reserved	Set to zero on Write. ignore on Read.	R
b7	discard	If set, BPDU frames are discarded always. Setting has no effect, when the enable bit is set.	R/W
b6	enable	If set, all Bridge Protocol Frames (BPDU) are forwarded exclusively to the management port specified in bits 3:0. BPDU frames received at the specified management port will be forwarded to the ports given in the portmask given in this register, bypassing the normal forwarding decisions (except forced forwarding). If cleared, Bridge Protocol Frames (BPDU) are forwarded as any other frame, or discarded if the discard bit is set.	R/W
b5	message_transmitted	Set (latched) when a BPDU message was transmitted from the management port to any output port. It can be used for handshaking to indicate that the portmask bits above were used and can now be changed again; e.g. set it to 0. Will not assert for normal frames. The Bit is reset by writing into the register.	R/W
b4	Reserved	Set to zero on Write. ignore on Read.	R
b3 to b0	Port	The Port number of the port that should act as a management port. Keep the initial value.	R/W

4.4.10 MODE_CONFIG — Mode Configuration Register

Address: 4405 0024h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	StatsReset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	Cut_Through_Enable					Options							
Value after reset	X	X	X	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.12 MODE_CONFIG Register Contents

Bit Position	Bit Name	Function	R/W
b31	StatsReset	Reset Statistics Counters Command. When set during a write, all statistics counters are cleared. When set, all other bits of the write are ignored and do not influence the currently stored value in the register (i.e. it is not necessary to read/preserve the register contents prior to writing the command). This also means the other bits of the register can be written only if bit31=0.	W
b30 to b13	Reserved	Set to zero on Write. ignore on Read.	R
b12 to b8	Cut_Through_Enable	Port Cut through Support Enable One bit per port. Bit 8 = Port 0, Bit 9 = Port 1, Bit 10 = Port 2, Bit 11 = Port 3, Bit 12 = Port 4. When a port is enabled for cut through, a frame received is forwarded to the destination before the completion of the frame reception. Cut Through is possible only if both ports (where frame is received and where frame is forwarded to) have the cut through bit set. Otherwise the frame is forwarded in normal store & forward behavior.	R/W
b7 to b0	Options	Reserved, unused. Write 0 always.	R

4.4.11 VLAN_IN_MODE — VLAN Input Manipulation Mode Register

Address: 4405 0028h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	P4VLANINMD		P3VLANINMD		P2VLANINMD		P1VLANINMD		P0VLANINMD	
Value after reset	X	X	X	X	X	X	0	0	0	0	0	0	0	0	0	0

Table 4.13 VLAN_IN_MODE Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b10	Reserved	Set to zero on Write. ignore on Read.	R
b9, b8	P4VLANINMD	Port4 Define Behavior of VLAN Input Manipulation Function 00b: Mode 1 01b: Mode 2 10b: Mode 3 11b: Mode 4	R/W
b7, b6	P3VLANINMD	Port3 Define Behavior of VLAN Input Manipulation Function 00b: Mode 1 01b: Mode 2 10b: Mode 3 11b: Mode 4	R/W
b5, b4	P2VLANINMD	Port2 Define Behavior of VLAN Input Manipulation Function 00b: Mode 1 01b: Mode 2 10b: Mode 3 11b: Mode 4	R/W
b3, b2	P1VLANINMD	Port1 Define Behavior of VLAN Input Manipulation Function 00b: Mode 1 01b: Mode 2 10b: Mode 3 11b: Mode 4	R/W
b1, b0	P0VLANINMD	Port0 Define Behavior of VLAN Input Manipulation Function 00b: Mode 1 01b: Mode 2 10b: Mode 3 11b: Mode 4	R/W

4.4.12 VLAN_OUT_MODE — VLAN Output Manipulation Mode Register

Address: 4405 002Ch

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	P4VLANOUTMD		P3VLANOUTMD		P2VLANOUTMD		P1VLANOUTMD		P0VLANOUTMD	
Value after reset	X	X	X	X	X	X	0	0	0	0	0	0	0	0	0	0

Table 4.14 VLAN_OUT_MODE Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b10	Reserved	Set to zero on Write. ignore on Read.	R
b9, b8	P4VLANOUTMD	Port4 Define Behavior of VLAN Output Manipulation Function 00b: No output manipulation 01b: Mode 1, Strip Mode 10b: Mode 2, Tag Through 11b: Mode 3, Transparent	R/W
b7, b6	P3VLANOUTMD	Port3 Define Behavior of VLAN Output Manipulation Function 00b: No output manipulation 01b: Mode 1, Strip Mode 10b: Mode 2, Tag Through 11b: Mode 3, Transparent	R/W
b5, b4	P2VLANOUTMD	Port2 Define Behavior of VLAN Output Manipulation Function 00b: No output manipulation 01b: Mode 1, Strip Mode 10b: Mode 2, Tag Through 11b: Mode 3, Transparent	R/W
b3, b2	P1VLANOUTMD	Port1 Define Behavior of VLAN Output Manipulation Function 00b: No output manipulation 01b: Mode 1, Strip Mode 10b: Mode 2, Tag Through 11b: Mode 3, Transparent	R/W
b1, b0	P0VLANOUTMD	Port0 Define Behavior of VLAN Output Manipulation Function 00b: No output manipulation 01b: Mode 1, Strip Mode 10b: Mode 2, Tag Through 11b: Mode 3, Transparent	R/W

4.4.13 VLAN_IN_MODE_ENA — VLAN Input Mode Enable Register

Enable the input processing according to the VLAN_IN_MODE for a port (1 bit per port). When disabled (bit = 0), the VLAN_IN_MODE setting for that port has no effect and the frames are processed unmodified.

Address: 4405 0030h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	VLANINMDEN				
Value after reset	X	X	X	X	X	X	X	X	X	X	X	0	0	0	0	0

Table 4.15 VLAN_IN_MODE_ENA Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b5	Reserved	Set to zero on Write. ignore on Read.	R
b4 to b0	VLANINMDEN	Enable the input processing according to the VLAN_IN_MODE for a port (1 bit per port).	R/W

4.4.14 VLAN_TAG_ID — VLAN Tag ID Register

Address: 4405 0034h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	VLANTAGID															
Value after reset	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

Table 4.16 VLAN_TAG_ID Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b16	Reserved	Set to zero on Write. ignore on Read.	R
b15 to b0	VLANTAGID	The VLAN type field (TPID) value to expect to identify a VLAN tagged frame. Default value is 0x8100.	R

4.4.15 BCAST_STORM_LIMIT — Broadcast Storm Protection Register

Address: 4405 0038h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	BCASTLIMIT															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TMOUT															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.17 BCAST_STORM_LIMIT Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b16	BCASTLIMIT	Number of broadcast frames (-1) that can be accepted on a port during a timeout period. If more are received, they are discarded. The counter is implemented per port independently; however the limit is used for all ports. When 0, no limit is set. Value is set one less. (i.e. set 9 to allow 10 frames)	R/W
b15 to b0	TMOUT	Timeout in steps of 65536 switch system clock cycles.	R/W

4.4.16 MCAST_STORM_LIMIT — Multicast Storm Protection Register

Address: 4405 003Ch

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	MCASTLIMIT															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Table 4.18 MCAST_STORM_LIMIT Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b16	MCASTLIMIT	Number of multicast frames (-1) that can be accepted on a port during a timeout period. If more are received, they are discarded. The counter is implemented per port independently; however the limit is used for all ports. When 0, no limit is set.	R/W
b15 to b0	Reserved	Set to zero on Write. ignore on Read.	R/W

4.4.17 MIRROR_CONTROL — Port Mirroring Configuration Register

Address: 4405 0040h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	eg_da_match	eg_sa_match	ing_da_match	ing_sa_match	eg_map_enable	ing_map_enable	mirror_enable	Mirror_Port			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.19 MIRROR_CONTROL Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b11	Reserved	Set to zero on Write. ignore on Read.	R
b10	eg_da_match	If set, only frames transmitted on an egress port with a destination address matching the value programmed in register MIRROR_EDST are mirrored. Other frames are not mirrored.	R/W
b9	eg_sa_match	If set, only frames transmitted on an egress port with a source address matching the value programmed in register MIRROR_ESRC are mirrored. Other frames are not mirrored.	R/W
b8	ing_da_match	If set, only frames received on an ingress port with a destination address matching the value programmed in register MIRROR_IDST are mirrored. Other frames are not mirrored.	R/W
b7	ing_sa_match	If set, only frames received on an ingress port with a source address matching the value programmed in register MIRROR_ISRC are mirrored. Other frames are not mirrored.	R/W
b6	eg_map_enable	If set, the egress map is enabled (MIRROR_EG_MAP).	R/W
b5	ing_map_enable	If set, the ingress map is enabled (MIRROR_ING_MAP).	R/W
b4	mirror_enable	0: Disable mirroring 1: Enable mirroring	R/W
b3 to b0	Mirror_Port	The port number of the port that should act as the mirror port and receive all mirrored frames. Valid setting range is 0 to 4.	R/W

4.4.18 MIRROR_EG_MAP — Port Mirroring Egress Port Definition

Address: 4405 0044h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	EMAP				
Value after reset	X	X	X	X	X	X	X	X	X	X	X	0	0	0	0	0

Table 4.20 MIRROR_EG_MAP Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b5	Reserved	Set to zero on Write. ignore on Read.	R
b4 to b0	EMAP	Port Mirroring Egress Port Definitions 1 bit per port. If enabled (Bit = 1), frames destined to the port(s) are mirrored to the mirror port	R/W

4.4.19 MIRROR_ING_MAP — Port Mirroring Ingress Port Definition

Address: 4405 0048h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	IMAP				
Value after reset	X	X	X	X	X	X	X	X	X	X	X	0	0	0	0	0

Table 4.21 MIRROR_ING_MAP Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b5	Reserved	Set to zero on Write. ignore on Read.	R
b4 to b0	IMAP	Port Mirroring Ingress Port Definitions 1 bit per port. If enabled (Bit = 1), frames from the port(s) are mirrored on the mirror port	R/W

4.4.20 MIRROR_ISRC_0 — Ingress Source MAC Address for Mirror Filtering 0

Address: 4405 004Ch

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	ISRC															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ISRC															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.22 MIRROR_ISRC_0 Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	ISRC	Ingress Source MAC Address for Mirror Filtering Lowest 32-bit of address.	R/W

4.4.21 MIRROR_ISRC_1 — Ingress Source MAC Address for Mirror Filtering 1

Address: 4405 0050h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ISRC															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.23 MIRROR_ISRC_1 Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b16	Reserved	Set to zero on Write. ignore on Read.	R
b15 to b0	ISRC	Ingress Source MAC Address for Mirror Filtering Upper 16-bit of address.	R/W

4.4.22 MIRROR_IDST_0 — Ingress Destination MAC Address for Mirror Filtering 0

Address: 4405 0054h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	IDST															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	IDST															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.24 MIRROR_IDST_0 Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	IDST	Ingress Destination MAC Address for Mirror Filtering Lowest 32-bit of address.	R/W

4.4.23 MIRROR_IDST_1 — Ingress Destination MAC Address for Mirror Filtering 1

Address: 4405 0058h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	IDST															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.25 MIRROR_IDST_1 Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b16	Reserved	Set to zero on Write. ignore on Read.	R
b15 to b0	IDST	Ingress Destination MAC Address for Mirror Filtering Upper 16-bit of address.	R/W

4.4.24 MIRROR_ESRC_0 — Egress Source MAC Address for Mirror Filtering 0

Address: 4405 005Ch

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	ESRC															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ESRC															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.26 MIRROR_ESRC_0 Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	ESRC	Egress Source MAC Address for Mirror Filtering Lowest 32-bit of address.	R/W

4.4.25 MIRROR_ESRC_1 — Egress Source MAC Address for Mirror Filtering 1

Address: 4405 0060h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ESRC															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.27 MIRROR_ESRC_1 Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b16	Reserved	Set to zero on Write. ignore on Read.	R
b15 to b0	ESRC	Egress Source MAC Address for Mirror Filtering Upper 16-bit of address.	R/W

4.4.26 MIRROR_EDST_0 — Egress Destination MAC Address for Mirror Filtering 0

Address: 4405 0064h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	EDST															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	EDST															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.28 MIRROR_EDST_0 Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	EDST	Egress Destination MAC Address for Mirror Filtering Lowest 32-bit of address.	R/W

4.4.27 MIRROR_EDST_1 — Egress Destination MAC Address for Mirror Filtering 1

Address: 4405 0068h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	EDST															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.29 MIRROR_EDST_1 Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b16	Reserved	Set to zero on Write. ignore on Read.	R
b15 to b0	EDST	Egress Destination MAC Address for Mirror Filtering Upper 16-bit of address.	R/W

4.4.28 MIRROR_CNT — Mirror Filtering Count Value Register

Address: 4405 006Ch

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	CNT							
Value after reset	X	X	X	X	X	X	X	X	0	0	0	0	0	0	0	0

Table 4.30 MIRROR_CNT Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b8	Reserved	Set to zero on Write. ignore on Read.	R
b7 to b0	CNT	Count Value for Mirror Filtering Every n-th frame is forwarded to the mirror port if enabled. A value of 1 or 0 means every frame.	R/W

4.4.29 QMGR_ST_MINCELLS — Output Queue Minimum Memory Statistics Register

Address: 4405 0088h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	M_CELLS_MIN										
Value after reset	X	X	X	X	X	0	0	0	0	0	0	0	0	0	0	0

Table 4.31 QMGR_ST_MINCELLS Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b11	Reserved	Set to zero on Write. ignore on Read.	R
b10 to b0	M_CELLS_MIN	Statistic providing the lowest number of free cells reached in memory during operation since this statistic was last cleared. The value is reset to the maximum when a write to the register with any value is performed.	R/W

4.4.30 QMGR_RED_MIN4 — RED Minimum Threshold Register

Address: 4405 0094h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	CFGRED_MINTH4															
Value after reset	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CFGRED_MINTH4															
Value after reset	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1

Table 4.32 QMGR_RED_MIN4 Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	CFGRED_MINTH4	Random Early Detection (RED) Minimum Threshold for Queues 0..3. An 8 bits value per queue, global for all ports: CFGRED_MINTH4[7:0]: queue 0 CFGRED_MINTH4[15:8]: queue 1 CFGRED_MINTH4[23:16]: queue 2 CFGRED_MINTH4[31:24]: queue 3	R/W

4.4.31 QMGR_RED_MAX4 — RED Maximum Threshold Register

Address: 4405 0098h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	CFGRED_MAXTH4															
Value after reset	0	0	1	0	1	1	1	1	0	0	1	0	1	1	1	1

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CFGRED_MAXTH4															
Value after reset	0	0	1	0	1	1	1	1	0	0	1	0	1	1	1	1

Table 4.33 QMGR_RED_MAX4 Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	CFGRED_MAXTH4	Random Early Detection (RED) Maximum Threshold for Queues 0..3. An 8 bits value per queue, global for all ports: CFGRED_MAXTH4[7:0]: queue 0 CFGRED_MAXTH4[15:8]: queue 1 CFGRED_MAXTH4[23:16]: queue 2 CFGRED_MAXTH4[31:24]: queue 3	R/W

4.4.32 QMGR_RED_CONFIG — RED Configuration Register

Address: 4405 009Ch

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	GACTIVITY_EN	—	—	—	—	QUEUE_RED_EN			
Value after reset	X	X	X	X	X	X	X	0	X	X	X	X	0	0	0	0

Table 4.34 QMGR_RED_CONFIG Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b9	Reserved	Set to zero on Write. ignore on Read.	R
b8	GACTIVITY_EN	Enable Averaging on Global Switch Activity (1) or on port local activity (0) only.	R/W
b7 to b4	Reserved	Set to zero on Write. ignore on Read.	R
b3 to b0	QUEUE_RED_EN	Enable Random Early Detection (RED) (1) or Taildrop (0) congestion management for a queue. One bit per queue: bit 0 = queue0, bit 1 = queue1, … Per queue setting, but global for all queues of all ports.	R/W

4.4.33 IMC_STATUS — Input Memory Controller Status Register

Register reads 0x08000400 when memory initialization completed.

Address: 4405 00A0h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	mem_full	de_init	de_error	cf_error	cells_available							
Value after reset	X	X	X	X	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	cells_available															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.35 IMC_STATUS Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b28	Reserved	Set to zero on Write. ignore on Read.	R
b27	mem_full	Latched indication that memory is or was full. Asserts when all cells of the memory have been allocated (memory congestion). As a result, frames will be/were discarded. This is not an error. ***Note*)** Bit is Read clear	R
b26	de_init	Asserts during memory initialization (deallocation module). Asserts after reset as long as memory initialization is ongoing and then clears when the memory is ready for normal operation. ***Note*)** The switch must not be enabled before the memory is ready for operation	R
b25	de_error	Deallocation error. Indication that memory cells could not be deallocated and are lost for further usage. This is a fatal error that must never occur during normal operation as it indicates memory leakage. ***Note*)** Bit is Read clear	R
b24	cf_error	Cell factory empty error. Indication that memory overflow occurred. There were more memory cells requested than available. This is a fatal error that must never occur during normal operation (may indicate memory leakage). ***Note*)** Bit is Read clear	R
b23 to b0	cells_available	Total number of memory cells (128-byte units) available in the shared memory (real time).	R

4.4.34 IMC_ERR_FULL — Input Port Memory Full and Truncation Indicator

Address: 4405 00A4h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	ipc_err_trunc				
Value after reset	X	X	X	X	X	X	X	X	X	X	X	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	ipc_err_full				
Value after reset	X	X	X	X	X	X	X	X	X	X	X	0	0	0	0	0

Table 4.36 IMC_ERR_FULL Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b21	Reserved	Set to zero on Write. ignore on Read.	R
b20 to b16	ipc_err_trunc	Memory became full while a frame was received and was partly written into memory. A frame has been truncated and discarded. Shows congestion occurred on an input to the shared memory. This field is cleared by read. One bit per port (bit 0 = port 0, bit 1= port 1, ...).	R
b15 to b5	Reserved	Set to zero on Write. ignore on Read.	R
b4 to b0	ipc_err_full	Memory was full at start of a frame reception. A frame has been discarded. Shows congestion occurred on an input to the shared memory. This field is cleared by read. One bit per port (bit 0 = port 0, bit 1 = port 1, ...).	R

4.4.35 IMC_ERR_IFACE — Input Port Memory Error Indicator

Address: 4405 00A8h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	wbuf_oflow				
Value after reset	X	X	X	X	X	X	X	X	X	X	X	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	ipc_err_iface				
Value after reset	X	X	X	X	X	X	X	X	X	X	X	0	0	0	0	0

Table 4.37 IMC_ERR_IFACE Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b21	Reserved	Set to zero on Write. ignore on Read.	R
b20 to b16	wbuf_oflow	Error indicating an overflow in the input write buffer to the memory controller (a small decoupling FIFO at every MAC RX). This must not occur during normal operation. If it occurs, the received frame is corrupted and marked as error to be discarded eventually by the memory controller. If it occurs during normal operation, it indicates the switch system frequency is too low (i.e. MAC rx rate is higher than memory write bandwidth). This field is cleared by read. One bit per port (bit 0 = port 0, bit 1 = port 1, ...).	R
b15 to b5	Reserved	Set to zero on Write. ignore on Read.	R
b4 to b0	ipc_err_iface	Error indication on memory input (receive from MAC) that a frame has been truncated and discarded. This may occur during normal operation if e.g. input rules would discard a frame after it has started being written into memory (e.g. destination port mask all zero or VLAN domain verification fails or source address not found discard is enabled and source was not found). It also indicates internal (fatal!) errors if the memory access failed. If none of the input rules is active and these bits would still assert, it indicates a too low switch system clock frequency (i.e. input rate from MAC is too high for the memory to be able to store the data). This field is cleared by read. One bit per port (bit 0 = port 0, bit 1 = port 1, ...).	R

4.4.36 IMC_ERR_QOFLOW — Output Port Queue Overflow Indicator

Address: 4405 00ACh

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	op_error				
Value after reset	X	X	X	X	X	X	X	X	X	X	X	0	0	0	0	0

Table 4.38 IMC_ERR_QOFLOW Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b5	Reserved	Set to zero on Write. ignore on Read.	R
b4 to b0	op_error	A frame could not be stored in an output queue of the port as the queue FIFO overflowed (i.e. write occurred into full fifo). The frame is ignored but stays stored in memory. Should never occur during normal operation. This is a fatal error as the memory, allocated by that frame is then never freed (i.e. resulting in memory leakage). It indicates an error in the congestion function which must normally prevent writing a frame into a queue when the queue is not capable of accepting it. This field is cleared by read. One bit per port (bit 0 = port 0, bit 1 = port 1, ...).	R

4.4.37 IMC_CONFIG — Input Memory Controller Configuration Register

Address: 4405 00B0h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	wfq_enable
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	0

Table 4.39 IMC_CONFIG Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b1	Reserved	Set to zero on Write. ignore on Read.	R
b0	wfq_enable	Enable weighted fair queuing (1) or strict priority (0, default) output queue scheduling. 4 queue implementation: When wfq is set, the output queues are weighted with factors 1,2,4,8 for queues 0,1,2,3 respectively (queue 3 is highest priority). A higher weight causes the queue to be served with higher priority. ***Note*)** Must be 0 when TDMA scheduling is used.	R/W

4.4.38 GPARSER[n] — [n]th Parser of 1st Block (n = 0..3)

Port Snooping function parser configuration.

Address: 4405 00C0h + 4h × n

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	cmp_mask_or	offset_plus2	compare16	ipprotocol	ipdata	skipvlan	valid	offset_DA	—	offset					
Value after reset	X	0	0	0	0	0	0	0	0	X	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	compare_value								mask_value2							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.40 GPARSER[n] Register Contents (1/2)

Bit Position	Bit Name	Function	R/W
b31	Reserved	Set to zero on Write. ignore on Read.	R
b30	cmp_mask_or	Use the mask byte (7:0) as a 2nd compare value. When set, the parser reports a match if the byte at given offset matches the compare value (15:8) or the mask value (7:0). Only usable when compare16 = 0.	R/W
b29	offset_plus2	Repeats the comparison at offset+2, if the comparison at offset failed. Only usable when compare16 = 1. This can be used to create a comparison for UDP or TCP port numbers allowing checking if the port number exists in the source or in the destination port number field.	R/W
b28	compare16	When set, the mask (7:0) is used as a value to perform a 16-bit compare: Register bits 15:8 represent the byte at the given offset and bits 7:0 the byte following at offset+1 which matches the network byte order for 16-bit fields (e.g. setting a compare value of 0x0800 and offset 0 matches IP frames). No mask is available in this mode. When cleared, only one byte of data is compared and the mask can be used to mask individual bits.	R/W
b27	ipprotocol	When set, the compare value is compared with the protocol field found within the IP header for both IPv4 and IPv6 frames. It implicitly acts as skipvlan=1 skipping any VLAN tags if present. The offset setting has no meaning and is ignored. If the bit is set, but the frame is not an IPv4/v6 frame the parser will report “no match” and not continue to inspect the frame. When cleared, the offset is used normally on all frames. ***Note)*** When this bit is set, the bits 25,26,28,29,30 have no meaning and should be set to 0. The mask (7:0) is applied hence should be 0xff to achieve an exact match.	R/W
b26	ipdata	When set, the offset starts with the first byte following an IP header if an IP frame is processed. The following fields are skipped: • any VLAN tags if present (implicitly acts as skipvlan = 1) • for IPv4 the header and any header options • for IPv6 only the base header. If the bit is set, it implicitly requires an IP frame. If the frame is not an IP frame, the parser will report “no match” and not continue to inspect the frame. When cleared, the offset starts with the first data following the MAC source address (skipping VLAN tags if allowed by “skipvlan”). ***Note)*** When this bit is set, the bit skipvlan has no meaning and can be set to any value.	R/W

Table 4.40 GPARSER[n] Register Contents (2/2)

Bit Position	Bit Name	Function	R/W
b25	skipvlan	When set, any optional VLAN tags found in the frame are skipped and the parser starts operating at the first byte following any VLAN tags. When cleared, the parser starts with the first byte following the source MAC address. ***Note)*** If 0, this means the first byte of the Type/length field is the first data for any compares, allowing to check for ether Types.	R/W
b24	valid	Indicate that this entry is valid (1) and should be used. When 0, the parser result always indicates “no match” and none of the other bits are relevant.	R/W
b23	offset_DA	When set, the offset starts counting from the first byte of the MAC destination address. ***Note)*** The skipvlan and ipdata and ipprotocol bits must be 0 when this bit is set.	R/W
b22	Reserved	Set to zero on Write. ignore on Read.	R
b21 to b16	offset	An offset in bytes where to find the data for comparison within the frame. The offset value starts at 0 to indicate the very first byte after offset start. The offset start can be either the frame’s type/length field (i.e. 0 = first byte of type/length field) or the payload following an IP header (see bit “ipdata” above). Valid values range from 0 to 60.	R/W
b15 to b8	compare_value	The value to compare the frame data with at the given offset.	R/W
b7 to b0	mask_value2	Mask for single byte compares or 2nd compare value (if bit 30 = 1) or least significant bits of a 16-bit compare value (if bit 28 = 1). When used as a mask (bit 28,30 = 0,0), the data from the frame is ANDed with this mask, then compared to the compare value. All bits having a “1” in the mask will be compared with the data in the frame. All bits having a “0” will be 0 for the compare, however this requires the compare value to have those bits also set to “0”.	R/W

4.4.39 GARITH[n] — Snoop Configuration for Arithmetic [n]th Stage of 1st Block (n = 0..3)

Address: 4405 00D0h + 4h × n

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	snoopmode		—	—	result_invert	operation
Value after reset	X	X	X	X	X	X	X	X	X	X	0	0	X	X	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	select_arith2	select_arith1	select_arith0	select_match				—	—	—	—	not_input			
Value after reset	X	0	0	0	0	0	0	0	X	X	X	X	0	0	0	0

Table 4.41 GARITH[n] Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b22	Reserved	Set to zero on Write. ignore on Read.	R
b21, b20	snoopmode	00b: Disabled, no snooping occurs (forward normally) 01b: Forward to designated management port only 10b: Forward normally and copy to management port 11b: Discard the frame ***Note*)** • If both arithmetic blocks (Arithmetic 0 and 1) match at the same time and they have different mode settings, the lower mode will be chosen (e.g. if one function is programmed to 01b and the other to 11b then mode 01b will be executed). • These bits exist only in the GARITH3 register. The bits are not writeable in all others.	R/W
b19, b18	Reserved	Set to zero on Write. ignore on Read.	R
b17	result_invert	0: The output is used directly 1: The output of this stage is inverted	R/W
b16	operation	0: AND all selected inputs 1: OR all selected inputs	R/W
b15	Reserved	Set to zero on Write. ignore on Read.	R
b14	select_arith2	Available only on stage 3. If set, the result from arithmetic stage 2 is selected in addition to any of the parser results.	R/W
b13	select_arith1	Available only on stages 2..3. If set, the result from arithmetic stage 1 is selected in addition to any of the parser results.	R/W
b12	select_arith0	Available only on stages 1..3. If set, the result from arithmetic stage 0 is selected in addition to any of the parser results.	R/W
b11 to b8	select_match	Define, which parser result is relevant at this stage. One bit per parser. Bit8 = parser0, ..., Bit11 = parser3 When a bit is “1”, the match result of the corresponding parser is used in this stage, otherwise it is ignored.	R/W
b7 to b4	Reserved	Set to zero on Write. ignore on Read.	R
b3 to b0	not_input	Define, which match result from a parser should be used inverted. One bit per parser. Bit0 = parser0, Bit1 = parser1, ..., Bit3 = parser3 When a bit is “1”, the match result of the corresponding parser is used inverted in this stage.	R/W

4.4.40 GPARSER[n] — [n−4]th Parser of 2nd Block (n = 4..7)

Port Snooping function parser configuration.

Address: 4405 00E0h + 4h × (n − 4)

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	cmp_mask_or	offset_plus2	compare16	ipprotocol	ipdata	skipvlan	valid	offset_DA	—	offset					
Value after reset	X	0	0	0	0	0	0	0	0	X	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	compare_value								mask_value2							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.42 GPARSER[n] Register Contents (1/2)

Bit Position	Bit Name	Function	R/W
b31	Reserved	Set to zero on Write. ignore on Read.	R
b30	cmp_mask_or	Use the mask byte (7:0) as a 2nd compare value. When set, the parser reports a match if the byte at given offset matches the compare value (15:8) or the mask value (7:0). Only usable when compare16 = 0.	R/W
b29	offset_plus2	Repeats the comparison at offset+2, if the comparison at offset failed. Only usable when compare16 = 1. This can be used to create a comparison for UDP or TCP port numbers allowing checking if the port number exists in the source or in the destination port number field.	R/W
b28	compare16	When set, the mask (7:0) is used as a value to perform a 16-bit compare: Register bits 15:8 represent the byte at the given offset and bits 7:0 the byte following at offset+1 which matches the network byte order for 16-bit fields (e.g. setting a compare value of 0x0800 and offset 0 matches IP frames). No mask is available in this mode. When cleared, only one byte of data is compared and the mask can be used to mask individual bits.	R/W
b27	ipprotocol	When set, the compare value is compared with the protocol field found within the IP header for both IPv4 and IPv6 frames. It implicitly acts as skipvlan = 1 skipping any VLAN tags if present. The offset setting has no meaning and is ignored. If the bit is set, but the frame is not an IPv4/v6 frame the parser will report “no match” and not continue to inspect the frame. When cleared, the offset is used normally on all frames. ***Note)*** When this bit is set, the bits 25,26,28,29,30 have no meaning and should be set to 0. The mask (7:0) is applied hence should be 0xff to achieve an exact match.	R/W
b26	ipdata	When set, the offset starts with the first byte following an IP header if an IP frame is processed. The following fields are skipped: • any VLAN tags if present (implicitly acts as skipvlan = 1) • for IPv4 the header and any header options • for IPv6 only the base header. If the bit is set, it implicitly requires an IP frame. If the frame is not an IP frame, the parser will report “no match” and not continue to inspect the frame. When cleared, the offset starts with the first data following the MAC source address (skipping VLAN tags if allowed by “skipvlan”). ***Note)*** When this bit is set, the bit skipvlan has no meaning and can be set to any value.	R/W

Table 4.42 GPARSER[n] Register Contents (2/2)

Bit Position	Bit Name	Function	R/W
b25	skipvlan	When set, any optional VLAN tags found in the frame are skipped and the parser starts operating at the first byte following any VLAN tags. When cleared, the parser starts with the first byte following the source MAC address. ***Note)*** If 0, this means the first byte of the Type/length field is the first data for any compares, allowing to check for ether Types.	R/W
b24	valid	Indicate that this entry is valid (1) and should be used. When 0, the parser result always indicates “no match” and none of the other bits are relevant.	R/W
b23	offset_DA	When set, the offset starts counting from the first byte of the MAC destination address. ***Note)*** The skipvlan and ipdata and ipprotocol bits must be 0 when this bit is set.	R/W
b22	Reserved	Set to zero on Write. ignore on Read.	R
b21 to b16	offset	An offset in bytes where to find the data for comparison within the frame. The offset value starts at 0 to indicate the very first byte after offset start. The offset start can be either the frame’s type/length field (i.e. 0 = first byte of type/length field) or the payload following an IP header (see bit “ipdata” above). Valid values range from 0 to 60.	R/W
b15 to b8	compare_value	The value to compare the frame data with at the given offset.	R/W
b7 to b0	mask_value2	Mask for single byte compares or 2nd compare value (if bit30 = 1) or least significant bits of a 16-bit compare value (if bit28 = 1). When used as a mask (bit28, 30 = 0,0), the data from the frame is ANDed with this mask, then compared to the compare value. All bits having a “1” in the mask will be compared with the data in the frame. All bits having a “0” will be 0 for the compare, however this requires the compare value to have those bits also set to “0”.	R/W

4.4.41 GARITH[n] — Snoop Configuration for Arithmetic [n−4]th Stage of 2nd Block (n = 4..7)

Address: 4405 00F0h + 4h × (n − 4)

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	snoopmode		—	—	result_invert	operation
Value after reset	X	X	X	X	X	X	X	X	X	X	0	0	X	X	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	select_arith2	select_arith1	select_arith0	select_match				—	—	—	—	not_input			
Value after reset	X	0	0	0	0	0	0	0	X	X	X	X	0	0	0	0

Table 4.43 GARITH[n] Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b22	Reserved	Set to zero on Write. ignore on Read.	R
b21, b20	snoopmode	00b: Disabled, no snooping occurs (forward normally) 01b: Forward to designated management port only 10b: Forward normally and copy to management port 11b: Discard the frame ***Note*)** • If both arithmetic blocks (Arithmetic 0 and 1) match at the same time and they have different mode settings, the lower mode will be chosen (e.g. if one function is programmed to 01b and the other to 11b then mode 01b will be executed). • These bits exist only in the GARITH7 register. The bits are not writeable in all others.	R/W
b19, b18	Reserved	Set to zero on Write. ignore on Read.	R
b17	result_invert	0: The output is used directly 1: The output of this stage is inverted	R/W
b16	operation	0: AND all selected inputs 1: OR all selected inputs	R/W
b15	Reserved	Set to zero on Write. ignore on Read.	R
b14	select_arith2	Available only on stage 3. If set, the result from arithmetic stage 2 is selected in addition to any of the parser results.	R/W
b13	select_arith1	Available only on stages 2..3. If set, the result from arithmetic stage 1 is selected in addition to any of the parser results.	R/W
b12	select_arith0	Available only on stages 1..3. If set, the result from arithmetic stage 0 is selected in addition to any of the parser results.	R/W
b11 to b8	select_match	Define, which parser result is relevant at this stage. One bit per parser. Bit 8 = parser 0, ..., Bit 11 = parser 3 When a bit is “1”, the match result of the corresponding parser is used in this stage, otherwise it is ignored.	R/W
b7 to b4	Reserved	Set to zero on Write. ignore on Read.	R
b3 to b0	not_input	Define, which match result from a parser should be used inverted. One bit per parser. Bit 0 = parser 4, Bit 1 = parser 5, ..., Bit 3 = parser 7 When a bit is “1”, the match result of the corresponding parser is used inverted in this stage.	R/W

4.4.42 VLAN_PRIORITY[n] — VLAN Priority Register [n] (n = 0..4)

The VLAN_PRIORITY[n] registers implement 3 bits to 2 bits VLAN priority mapping capability.

For each port, one register is provided. The current frame's 3 bits VLAN priority field is used as an index and the corresponding priority is taken from the respective position of the register giving the final classification for the frame.

Priority 0 is the lowest priority.

Address: 4405 0100h + 4h × n

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	PRIOIN7		—	PRIOIN6		—	PRIOIN5
Value after reset	X	X	X	X	X	X	X	X	X	1	1	X	1	1	X	1

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PRIOIN5	—	PRIOIN4		—	PRIOIN3		—	PRIOIN2		—	PRIOIN1		—	PRIOIN0	
Value after reset	1	X	1	1	X	1	1	X	1	0	X	0	1	X	0	0

Table 4.44 VLAN_PRIORITY[n] Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b23	Reserved	Set to zero on Write. ignore on Read.	R
b22, b21	PRIOIN7	This field is used as priority queue number if the frame's 3 bits VLAN priority field is 7.	R/W
b20	Reserved	Set to zero on Write. ignore on Read.	R
b19, b18	PRIOIN6	This field is used as priority queue number if the frame's 3 bits VLAN priority field is 6.	R/W
b17	Reserved	Set to zero on Write. ignore on Read.	R
b16, b15	PRIOIN5	This field is used as priority queue number if the frame's 3 bits VLAN priority field is 5.	R/W
b14	Reserved	Set to zero on Write. ignore on Read.	R
b13, b12	PRIOIN4	This field is used as priority queue number if the frame's 3 bits VLAN priority field is 4.	R/W
b11	Reserved	Set to zero on Write. ignore on Read.	R
b10, b9	PRIOIN3	This field is used as priority queue number if the frame's 3 bits VLAN priority field is 3.	R/W
b8	Reserved	Set to zero on Write. ignore on Read.	R
b7, b6	PRIOIN2	This field is used as priority queue number if the frame's 3 bits VLAN priority field is 2.	R/W
b5	Reserved	Set to zero on Write. ignore on Read.	R
b4, b3	PRIOIN1	This field is used as priority queue number if the frame's 3 bits VLAN priority field is 1.	R/W
b2	Reserved	Set to zero on Write. ignore on Read.	R
b1, b0	PRIOIN0	This field is used as priority queue number if the frame's 3 bits VLAN priority field is 0.	R/W

4.4.43 IP_PRIORITY[n] — IP Priority Register [n] (n = 0..4)

Address: 4405 0140h + 4h × n

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	read	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	priority		IPv6_select	Address							
Value after reset	X	X	X	X	X	0	0	0	0	0	0	0	0	0	0	0

Table 4.45 IP_PRIORITY[n] Register Contents

Bit Position	Bit Name	Function	R/W
b31	read	Must be cleared to write values in the tables. When set during register writes, the IPv6 select and address bits are stored in the register only and the priority bits are ignored and not written into the addressed table. When the register is read, the priority bits represent the value read from the table always.	R/W
b30 to b11	Reserved	Set to zero on Write. ignore on Read.	R
b10, b9	priority	The priority information to write into the addressed table entry. When reading from the register, the bits show the value from the addressed table entry (address from last write operation).	R/W
b8	IPv6_select	If set during a write, the IPv6 table is accessed. Valid address values range from 0 to 255. If cleared, the IPv4 table is accessed. Valid address values range from 0 to 63.	R/W
b7 to b0	Address	The address of the priority entry to read or write for a frame received on port [n]. The IPv4 table has 64 entries. The IPv6 table has 256 entries.	R/W

4.4.44 PRIORITY_CFG[n] — Priority Configuration Register [n] (n = 0..4)

Address: 4405 0180h + 4h × n

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	default_priority		TYPE_en	MAC_en	IP_en	VLAN_en
Value after reset	X	X	X	X	X	X	X	X	X	X	0	0	0	0	0	0

Table 4.46 PRIORITY_CFG[n] Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b6	Reserved	Set to zero on Write. ignore on Read.	R
b5, b4	default_priority	The default priority of a frame received on port [n], if none of the priority resolutions could define a priority of the frame.	R/W
b3	TYPE_en	Enable TYPE Based Priority Resolution for Frame Received on Port[n] If set, the frame's type/length field (following any VLAN tags) is compared to all PRIORITY_TYPE configured values. If any of the values matches, the priority information from that matching type will be used. If cleared, type fields are ignored.	R/W
b2	MAC_en	Enable MAC Based Priority Resolution for Frame Received on Port[n] If set, the priority information found within the MAC address table (for a static entry) is used. If cleared, MAC priority is ignored.	R/W
b1	IP_en	Enable IP Priority Resolution for Frame Received on Port[n] If set, the IP DiffServ/TrafficClass field is used and priority is resolved according to the IP_PRIORITY[n] setting for the port. If cleared, IP DiffServ/TrafficClass fields are ignored.	R/W
b0	VLAN_en	Enable VLAN Priority Resolution for Frame Received on Port[n] If set, the VLAN tag field of a frame is inspected and priority is resolved according to the setting programmed in VLAN_PRIORITY[n] for the port on which the frame was received. If cleared, VLAN priority is ignored.	R/W

4.4.45 PRIORITY_TYPE1 — Priority Type Register 1

Address: 4405 01B8h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	priority		valid
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	typevalue															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.47 PRIORITY_TYPE1 Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b19	Reserved	Set to zero on Write. ignore on Read.	R
b18, b17	priority	The priority value to use if a match occurs. ***Note*)** A port's PRIORITY_CFG[n].TYPE_en bit controls use of this priority.	R/W
b16	valid	If set indicates, this register contains valid data. If cleared (default) the data from this register is ignored. ***Note*)** When using with the Management port special frame tagging the valid bit has no relevance. The type is compared if enabled in the MGMT_TAG_CONFIG register. This allows using the type for the management tag only, or at the same time for priority resolution (e.g. useful for 1588 frames).	R/W
b15 to b0	typevalue	A 16-bit value to be compared against the frame's type/length field at receive. (e.g. a value of 0x88f7 would match IEEE 1588 frames). All such PRIORITY_TYPE registers are searched simultaneously and a type match will occur if any compare is successful. The value is relevant (and used for comparison) only if the valid bit is set. ***Note*)** This value is always compared against the type/length field of the frame following any VLAN tags. Up to two VLAN tags can be present in the frame. When used with the management port special frame tagging (see MGMT_TAG_CONFIG register), no VLAN tags can be supported for the tag insertion decision. That is, the management port tag will be inserted only if this is the first tag found in the frame. At the same time, for priority definition, VLAN tags are supported.	R/W

4.4.46 PRIORITY_TYPE2 — Priority Type Register 2

Address: 4405 01BCh

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	priority		valid
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	typevalue															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.48 PRIORITY_TYPE2 Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b19	Reserved	Set to zero on Write. ignore on Read.	R
b18, b17	priority	The priority value to use if a match occurs. ***Note*)** A port's PRIORITY_CFG[n].TYPE_en bit controls use of this priority.	R/W
b16	valid	If set indicates, this register contains valid data. ***Note*)** When using with the Management port special frame tagging the valid bit has no relevance. The type is compared if enabled in the MGMT_TAG_CONFIG register. This allows using the type for the management tag only, or at the same time for priority resolution (e.g. useful for IEEE 1588 frames).	R/W
b15 to b0	typevalue	A 16-bit value to be compared against the frame's type/length field at receive. (e.g. a value of 0x88f7 would match IEEE 1588 frames). All such PRIORITY_TYPE registers are searched simultaneously and a type match will occur if any compare is successful. The value is relevant (and used for comparison) only if the valid bit is set. ***Note*)** This value is always compared against the type/length field of the frame following any VLAN tags. Up to two VLAN tags can be present in the frame. When used with the management port special frame tagging (see MGMT_TAG_CONFIG register), no VLAN tags can be supported for the tag insertion decision. That is, the management port tag will be inserted only if this is the first tag found in the frame. At the same time, for priority definition, VLAN tags are supported.	R/W

4.4.47 MGMT_ADDR0_lo — Lower MAC Address for Bridge Protocol Frame

First 4 bytes of MAC address

Address: 4405 01C0h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	bpdu_dst_custom															
Value after reset	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	bpdu_dst_custom															
Value after reset	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Table 4.49 MGMT_ADDR0_lo Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	bpdu_dst_custom	Additional MAC address defining a Bridge Protocol Frame (BPDU) in addition to the well-known addresses First 4 bytes of MAC address: Bits 7:0 = 1st, 15:8 = 2nd, 23:16 = 3rd, 31:24 = 4th.	R/W

4.4.48 MGMT_ADDR0_hi — Higher MAC Address for Bridge Protocol Frame

Last 2 octets of MAC address and mask/valid bits.

NOTE

The compare is active always. To disable it, set a default value. Reset defaults to 01-80-c2-00-00-00 mask ffh.

Address: 4405 01C4h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	MASK8BIT							
Value after reset	X	X	X	X	X	X	X	X	1	1	1	1	1	1	1	1

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	bpdu_dst_custom															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.50 MGMT_ADDR0_hi Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b24	Reserved	Set to zero on Write. ignore on Read.	R
b23 to b16	MASK8BIT	8-bit mask for comparing the last byte of the MAC address. Each bit being “1” in the mask will be compared. The mask is ANDed with the last byte of the MAC address (6th byte in frame) before comparison. Setting the mask to 0xff compares all bits. A mask of 0x00 will ignore the last byte of the MAC address and a match will occur on any value. Note that all mask bits with 0 must also have 0 in the corresponding MAC address bit (bits 15:8 in this register).	R/W
b15 to b0	bpdu_dst_custom	Bits 7:0 is 5th byte, Bits 15:8 is 6th (last) byte.	R/W

4.4.49 SYSTEM_TAGINFO[n] — One VLAN ID Field for VLAN Input Manipulation (n = 0..4)

Address: 4405 0200h + 4h × n

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SYSVLANINFO															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.51 SYSTEM_TAGINFO[n] Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b16	Reserved	Set to zero on Write. ignore on Read.	R
b15 to b0	SYSVLANINFO	System VLAN Info (prio/cfi/vid) for the Port [n]	R/W

4.4.50 AUTH_PORT[n] — PORT[n] Authentication Control and Configuration (n = 0..4)

The enable bits (both per port bits) reset defaults are controlled by STRAP_SX_ENB bit and STRAP_HUB_ENB bit of SWCTRL register.

STRAP_SX_ENB = 0, STRAP_HUB_ENB = 0: AUTH_PORT[4:0].authorized = 0h.
STRAP_SX_ENB = 0, STRAP_HUB_ENB = 1: AUTH_PORT[4:0].authorized = 3h.
STRAP_SX_ENB = 1, STRAP_HUB_ENB = 0: AUTH_PORT[4:0].authorized = fh.
STRAP_SX_ENB = 1, STRAP_HUB_ENB = 1: AUTH_PORT[4:0].authorized = 1fh.
STRAP_SX_ENB and STRAP_HUB_ENB are described in [**Section 10.2.14, SWCTRL — A5PSW Control Register**].

Address: 4405 0240h + 4h × n

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	guest_mask				
Value after reset	X	X	X	X	X	X	X	X	X	X	X	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	EAPOL_port_number				Auto_Change_Unauthorized	—	—	—	—	—	EAPOL_unicast_enable	BPDU_enable	guest_enable	EAPOL_enable	controlled_both	authorized
Value after reset	0	1	0	0	0	X	X	X	X	X	0	0	0	0	0	X

Table 4.52 AUTH_PORT[n] Register Contents (1/2)

Bit Position	Bit Name	Function	R/W
b31 to b21	Reserved	Set to zero on Write. ignore on Read.	R
b20 to b16	guest_mask	Destination port mask with all ports that are allowed to receive non EAPOL frames from this port while it is unauthorized and guest is enabled. Bit 16 = port 0, Bit 17 = port 1, … ***Note*)** • If all 0 frames are discarded even if guest is enabled. • If a destination port in this list is unauthorized and has its controlled both bit set, it is removed from this list automatically during forwarding. • If a destination port in this list is authorized it is not removed from this list. This allows forwarding of a frame from an unauthorized to an authorized port. This can be intentional to e.g. implement DHCP with an external server connected at an authorized port. It is the responsibility of the application to ensure updating all guest masks of all ports when a port changes to authorized and then should no longer get frames forwarded to it from unauthorized ports.	R/W
b15 to b12	EAPOL_port_number	4 bits port number where to send EAPOL frames to. Typically the management port (port number = 4), but could be an external port if desired (with limitations, see below). The setting must be identical for all ports. ***Note*)** • It is not recommended to change this setting. If it is changed and a port changes to authorized without keeping the EAPOL enable bit 2 set, then EAPOL frames will be treated as normal BPDU frames. Hence an EAPOL log off message will then not reach the port configured here but will be forwarded to the management port. • Setting is relevant also when a port is authorized and matches this number. If this port receives frames, it always can forward to unauthorized ports, even if their controlled both is set.	R/W

Table 4.52 AUTH_PORT[n] Register Contents (2/2)

Bit Position	Bit Name	Function	R/W
b11	Auto_Change_Unauthorized	Enables (if 1) automatic port change to unauthorized. When enabled and a frame of unknown source is received at the port, the authorized bit of this register is cleared. The port hence changes into unauthorized state and it is up to the application to re-authenticate the port and eventually set the authorized bit again. Has an effect only if the port has learning disabled (see **Section 4.4.8, INPUT_LEARN_BLOCK — Input Learning Block Register**) and is configured to discard frames of unknown source (see **Section 4.4.61, LK_CTRL — Learning/Lookup Function Global Configuration Register**) ***Note)*** The interrupt (INT_CONFIG.LK_NEW_SRC) can be used to inform the application of this event..	R/W
b10 to b6	Reserved	Set to zero on Write. ignore on Read.	R
b5	EAPOL_unicast_enable	Normally EAPOL frames must use the dedicated PAE multicast destination address. However if this bit is set and a unicast destination address is found, the frame is accepted if it matches the MAC address of the port as configured in the MAC's ADDR registers. It then will be forwarded to the EAPOL destination port number given in bits 15:12 (see above).	R/W
b4	BPDU_enable	Enable (1) / Disable (0) Reception of BPDU Frames When enabled and the port is unauthorized, BPDU frames are forwarded to the management port normally. Otherwise BPDU frames will be discarded. ***Note)*** Although EAPOL frames are using BPDU multicast addresses, they will not be discarded by having this bit 0.	R/W
b3	guest_enable	Enable (1) / Disable (0) Reception of non EAPOL Frames When enabled and the port is unauthorized, such frames are forwarded only to the given guest port mask. ***Note)*** When guest is enabled, the normal VLAN manipulation function (VLAN insertion) can be used if needed.	R/W
b2	EAPOL_enable	Enable (1) / Disable (0) Reception of EAPOL Frames When enabled EAPOL frames are forwarded to the port given in EAPOL destination port number (see above). This is independent from authorized state. If disabled and the port is: • Unauthorized: then EAPOL frames are discarded. • Authorized: then EAPOL frames are treated as BPDU frames and forwarded to the management port. ***Note)*** Would be disabled to implement the authentication statemachine's HELD state where a port is unauthorized and is not allowed to even accept EAPOL frames.	R/W
b1	controlled_both	If 1, the port operates in Controlled Directions mode "both". That is other ports cannot transmit any traffic on this port, except the port specified in EAPOL destination port. If 0, the port operates in Controlled Directions mode "in". This allows frames from other ports can be forwarded to this port for transmission, but receive is still restricted. ***Note)*** Represents OperControlledDirections from **Section 4.6.2** (4), 6.5	R/W
b0	authorized	If 1, the port is in the authorized state and operates normally without restrictions. The other bits of this register, except 15:12, 2, are then not relevant. If 0, the port is unauthorized restricting forwarding according to the rules configured in this register.	R/W

4.4.51 VLAN_RES_TABLE[n] — 32 VLAN Domain Entries (n = 0..31)

Address: 4405 0280h + 4h × n

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	write_portmask	write_tagmask	read_tagmask	—	—	—	—	—	—	—	—	—	—	—	VLANid
Value after reset	X	0	0	0	X	X	X	X	X	X	X	X	X	X	X	1

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	VLANid											Portmask				
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Table 4.53 VLAN_RES_TABLE[n] Register Contents

Bit Position	Bit Name	Function	R/W
b31	Reserved	Set to zero on Write. ignore on Read.	R
b30	write_portmask	When this bit is set when writing into the register, the bits (4:0) are stored in the portmask for the domain only. The tagged bit mask is left unchanged. The VLAN id must be set accordingly during the write. ***Note*)** When bits 29,30 are both 0 or both 1 during a write, the bits (4:0) are written always into both the portmask and the tagged bit mask.	R/W
b29	write_tagmask	When this bit is set when writing into the register, the bits (4:0) are stored in the tagged bit mask for the domain only. The portmask is left unchanged. The VLAN id must be set accordingly during the write. The tagged bit mask allows to specify for each port if the frames leaving the port should be tagged (bit = 1) or untagged (bit = 0). If a port is untagged, frames leaving that port will have the VLAN tag removed. This tagged bit mask is interpreted by the output manipulation function on all ports that operate in output manipulation mode 3. ***Note*)** When bits 29,30 are both 0 or both 1 during a write, the bits (4:0) are written always into both the portmask and the tagged bit mask.	R/W
b28	read_tagmask	Select contents of mask bits (4:0) when reading the register. If this bit is set during a write into the register, all other bits of the write are ignored (i.e. 30,29,16:0) and the bit 28 of the register toggles (1-> 0; 0-> 1). This is used only to allow changing the bit 28 without changing any table contents. When the register is read, the bit indicates the contents of bits (4:0): if 0: the portmask is returned. if 1: the tagged bit mask is returned. The bit is cleared when the register is written with bit28 = 0.	R/W
b27 to b17	Reserved	Set to zero on Write. ignore on Read.	R
b16 to b5	VLANid	The 12-bit VLAN identifier of the entry.	R/W
b4 to b0	Portmask	A bit set 1 defines a port as being member of the VLAN. When bit28/29 is set, the tagged bit mask is read/written instead.	R/W

4.4.52 TOTAL_DISC — Discarded Frame Total Number Register

Address: 4405 0300h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TOTAL_DISC															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TOTAL_DISC															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.54 TOTAL_DISC Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	TOTAL_DISC	Total number of incoming frames accepted by MAC RX but discarded in the switch.	R

4.4.53 TOTAL_BYT_DISC — Discarded Frame Total Bytes Register

Address: 4405 0304h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TOTAL_BYT_DISC															
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TOTAL_BYT_DISC															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.55 TOTAL_BYT_DISC Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	TOTAL_BYT_DISC	Sum of bytes of frames counted in TOTAL_DISC.	R

4.4.54 TOTAL_FRM — Processed Frame Total Number Register

NOTE

Increments also when cut through forwarding is used and a frame with invalid CRC was received, as it has been forwarded. The MAC's ifOutError however will indicate erroneous frames forwarded.

Address: 4405 0308h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TOTAL_FRM															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TOTAL_FRM															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.56 TOTAL_FRM Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	TOTAL_FRM	Total number of incoming frames processed by switch.	R

4.4.55 TOTAL_BYT_FRM — Processed Frame Total Bytes Register

Address: 4405 030Ch

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TOTAL_BYT_FRM															
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TOTAL_BYT_FRM															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.57 TOTAL_BYT_FRM Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	TOTAL_BYT_FRM	Sum of bytes of frames counted in TOTAL_FRM.	R

4.4.56 ODISC[n] — PORT[n] Discarded Outgoing Frame Count Register (n = 0..4)

Address: 4405 0310h + 10h × n

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	ODISC[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ODISC[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.58 ODISC[n] Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	ODISC[n]	PORT[n] Outgoing frames discarded due to output Queue congestion.	R

4.4.57 IDISC_VLAN[n] — PORT[n] Discarded Incoming VLAN Tagged Frame Count Register (n = 0..4)

Address: 4405 0314h + 10h × n

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	IDISC_VLAN[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	IDISC_VLAN[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.59 IDISC_VLAN[n] Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	IDISC_VLAN[n]	PORT[n] incoming frames discarded due to mismatching or missing VLAN id while VLAN verification was enabled.	R

4.4.58 IDISC_UNTAGGED[n] — PORT[n] Discarded Incoming VLAN Untagged Frame Count Register (n = 0..4)

Address: 4405 0318h + 10h × n

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	IDISC_UNTAGGED[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	IDISC_UNTAGGED[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.60 IDISC_UNTAGGED[n] Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	IDISC_UNTAGGED[n]	PORT[n] incoming frames discarded due to missing VLAN tag.	R

4.4.59 IDISC_BLOCKED[n] — PORT[n] Discarded Incoming Blocked Frame Count Register (n = 0..4)

Address: 4405 031Ch + 10h × n

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	IDISC_BLOCKED[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	IDISC_BLOCKED[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.61 IDISC_BLOCKED[n] Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	IDISC_BLOCKED[n]	PORT[n] incoming frames discarded (after learning) as port is configured in blocking mode.	R

4.4.60 IMC_QLEVEL_P[n] — PORT[n] Queued Frame Count Register (n = 0..4)

Address: 4405 03C0h + 4h × n

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	Queue3				Queue2				Queue1				Queue0			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.62 IMC_QLEVEL_P[n] Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b16	Reserved	Set to zero on Write. ignore on Read.	R
b15 to b12	Queue3	A 4 bits value per queue indicating number of frames stored in the queue3. Saturates at 15 if more than 15 frames are stored in the queue.	R
b11 to b8	Queue2	A 4 bits value per queue indicating number of frames stored in the queue2. Saturates at 15 if more than 15 frames are stored in the queue.	R
b7 to b4	Queue1	A 4 bits value per queue indicating number of frames stored in the queue1. Saturates at 15 if more than 15 frames are stored in the queue.	R
b3 to b0	Queue0	A 4 bits value per queue indicating number of frames stored in the queue0. Saturates at 15 if more than 15 frames are stored in the queue.	R

4.4.61 LK_CTRL — Learning/Lookup Function Global Configuration Register

NOTE

Bits [3:0] would all be set to “1” for a normal switch operation.

Address: 4405 0400h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	Discard_Unknown_Source				
Value after reset	X	X	X	X	X	X	X	X	X	X	X	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	Clear_Table	—	Discard_Unknown_Destination	Allow_Migration	Enable_Aging	Enable_Learning	Enable_Lookup
Value after reset	X	X	X	X	X	X	X	X	X	0	X	0	1	1	1	1

Table 4.63 LK_CTRL Register Contents (1/2)

Bit Position	Bit Name	Function	R/W
b31 to b21	Reserved	Set to zero on Write. ignore on Read.	R
b20 to b16	Discard_Unknown_Source	Per port discard if source address not found. Bit 16 = port 0, Bit 17 = port 1, ... When enabled (1) and the source address of an incoming frame is not found, the frame is discarded. The source will not be learned, even if learning is allowed (INPUT_LEARN_BLOCK). Such discard events are counted in the switch per port input statistics IDISC_BLOCKED. When disabled (0), such frames are accepted normally. This setting does not affect BPDU frames, which will be accepted always. Independently from this setting, the unknown source interrupt will be triggered, and the new address is stored internally for retrieval with the LK_ADDR_CTRL.GETLASTNEW command. ***Note)*** This function can have a negative impact on performance for Gigabit ports when using a (too) low system frequency (can cause undesired short frame drop).	R/W
b15 to b7	Reserved	Set to zero on Write. ignore on Read.	R
b6	Clear_Table	Writes all table entries with 0. When the bit is set “1”, the bit stays 1 until the function has completed. Lookup is disabled during this time and the switch will flood all frames. The BUSY bit within register LK_ADDR_CTRL is also set until the function has completed. ***Note)*** The bit is set after reset to flush the table. Software may read the bit as 1 if the flushing is not completed and then should not enable the switch until the bit changes to 0.	R/W
b5	Reserved	Set to zero on Write. ignore on Read.	R
b4	Discard_Unknown_Destination	When enabled (1) and the destination address is not found, the frame is discarded. This setting does not affect BPDU frames, which will be accepted always. When disabled (0), such frames are flooded normally.	R/W
b3	Allow_Migration	When enabled (1), existing dynamic entries in the lookup table are updated if the source port of the entry changed. Requires learning (bit1) to be enabled. When disabled (0), no update occurs on dynamic entries when the incoming port number changes.	R/W
b2	Enable_Aging	When enabled (1), the aging process continuously scans the table for outdated entries and removes them. When disabled (0), no aging occurs.	R/W

Table 4.63 LK_CTRL Register Contents (2/2)

Bit Position	Bit Name	Function	R/W
b1	Enable_Learning	When enabled (1), a frame source address, if not found in the lookup memory will be added automatically by the hardware. The source lookup will be followed by a store operation, storing the MAC address in the next free address of the lookup memory. When disabled (0), unknown source addresses will not be added to the memory automatically and it is up to the software to maintain the table entries	R/W
b0	Enable_Lookup	Enable Lookup Controller The Lookup controller can be enabled or disabled by the host CPU. When disabled, the switch forwarding engine will skip the lookup and flood all incoming frames as if no address was found.	R/W

4.4.62 LK_STATUS — Status Bits and Table Overflow Counter

Address: 4405 0404h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	learnevent	—	overflows													
Value after reset	0	X	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ageaddress															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.64 LK_STATUS Register Contents

Bit Position	Bit Name	Function	R/W
b31	learnevent	Indicates a new source address was detected. The bit is set whenever the lookup task could not find the source address from a received frame in the MAC address table. Bit is cleared when writing a 1 to this bit, or when issuing the GETLASTNEW command in LK_ADDR_CTRL to retrieve the latest entry. The bit asserts for every newly learned source address, independent from automatic learning or any discard options being configured. If the bit is already set from a previous event, it is left set.	R/W
b30	Reserved	Set to zero on Write. ignore on Read.	R
b29 to b16	overflows	Counts number of table overflows that occurred (a new address was learned but the table had no storage and an older entry was deleted). The counter is cleared by writing into the register and having bit16 = 1.	R/W
b15 to b0	ageaddress	Address the aging process will inspect when the aging timer expires next time.	R

4.4.63 LK_ADDR_CTRL — Address Table Transaction Control and Read/Write Address

Address: 4405 0408h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	BUSY	DELETE_PORT	CLEAR	LOOKUP	WAIT_COMPLETE	READ	WRITE	GETLASTNEW	CLEAR_STATIC	CLEAR_DYNAMIC	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	address_mask												
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.65 LK_ADDR_CTRL Register Contents (1/2)

Bit Position	Bit Name	Function	R/W
b31	BUSY	Transaction Busy Indication Controller is busy. As long as the controller is busy, the corresponding command bit is set. When it becomes non busy again, all command bits are cleared.	R
b30	DELETE_PORT	Scans the complete table for valid dynamic entries that contain the given port(s) in their destination port mask and deletes the port(s) or the complete entry. The port mask is provided in the address_mask[12:0] when writing this register (1 bit per port, bit 0 = port 0, bit 1 = port 1, etc.). ***Note)*** The given port mask is ANDed with the port mask of each entry. When the result is non zero, the entry is processed, otherwise it is ignored. When an entry is processed, the bits given in the mask are cleared in the entry's port mask. If the resulting port mask is then all zero, the entry is deleted from the table (i.e. valid bit = 0). The function operates on dynamic entries only and does not affect static entries. The bit stays 1 together with the BUSY indication until the table has been scanned completely.	R/W
b29	CLEAR	Writes all zero to the entry selected by the given address. If set together with the LOOKUP, first a lookup is performed and if the lookup succeeds, the entry is then deleted. The registers LK_DATA_LO/HI will also be cleared. The memory address in this register will be set from the lookup result. If the lookup failed, the clear command is ignored (and memory address is arbitrary).	R/W
b28	LOOKUP	Performs a lookup of the MAC address given in LK_DATA_LO/HI. When the lookup completed, the LK_DATA_HI upper 16-bit are updated with the corresponding bits read from the found entry. The address_mask field in this register will show the address where the entry is found. If the MAC address was not found, LK_DATA_HI valid bit (bit16) will be zero and the bits 31..17 are arbitrary. The address field of this register is changed to the correct memory address that is empty and can be used to write the new entry to (i.e. it returns the hash calculated for the MAC address plus an offset to the empty entry within the 8 entries block). The bit stays 1 together with the BUSY indication until the function completed.	R/W
b27	WAIT_COMPLETE	Instructs to stall the processor bus until the transaction is completed. This allows performing consecutive writes into this register with varying commands without the need for polling the busy bit.	R/W
b26	READ	Perform Single Read Transaction Data is returned in LK_DATA_LO/HI.	R/W
b25	WRITE	Perform a Single Write Transaction LK_DATA_LO/HI must be set prior to starting the transactions.	R/W

Table 4.65 LK_ADDR_CTRL Register Contents (2/2)

Bit Position	Bit Name	Function	R/W
b24	GETLASTNEW	Retrieves the last source address that was not found in the table and places it into LK_DATA_LO/HI. The valid bit of the entry (LK_DATA_HI(16)) indicates if the address is new (1) or not (0) since the command was last issued. The port mask indicates at which port that new address was received. Whenever the learning task detects an unknown source address, it stores it into an internal storage that can be retrieved by this command. This occurs for every new source address, no matter if learning is enabled or not. It stores the address also when the no source discard feature is enabled. Only one entry is stored; If another new address is found before the data is read, it overwrites any existing data.	R/W
b23	CLEAR_STATIC	Scans the complete table for valid static entries and deletes them (writes entry with all zero). Bit is cleared when the function has completed.	R/W
b22	CLEAR_DYNAMIC	Scans the complete table for valid dynamic entries and deletes them (writes entry with all zero). Bit is cleared when the function has completed.	R/W
b21 to b13	Reserved	Set to zero on Write. ignore on Read.	R
b12 to b0	address_mask	Memory address for read and write transactions. This is the address of a 64-bit entry. For the DELETE_PORT function, a port mask can be provided in these bits instead of the address. Bit 0 represents port 0, Bit 1 port 1, etc. When a LOOKUP function was executed and it found the address (LK_DATA_HI(16) = 1), the memory address of the entry is returned. When a LOOKUP function was executed and it could not find the address (LK_DATA_HI(16) = 0), it returns the valid memory address of an empty entry which can be used to store the new data if needed (for a 8192 size table, bits 12:3 contain the calculated hash value, bits 2:0 the offset within the 8 entries block).	R/W

4.4.64 LK_DATA_LO — Lower 32-Bit Data of Lookup Memory Entry

Address: 4405 040Ch

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	MEMDATA															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MEMDATA															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.66 LK_DATA_LO Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	MEMDATA	Memory Data (31:0) The lower 32-bit data of a memory entry. When writing MAC addresses, the first byte is 7:0 and the 4th byte is 31:24. When reading, the data returned is the captured data of the last read transaction.	R/W

4.4.65 LK_DATA_HI — Higher 26-Bit Data of Lookup Memory Entry

Address: 4405 0410h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	MEMDATA									
Value after reset	X	X	X	X	X	X	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MEMDATA															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.67 LK_DATA_HI Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b26	Reserved	Set to zero on Write. ignore on Read.	R
b25 to b0	MEMDATA	Memory Data (57:32) The next 26-bit data of a memory entry. When writing MAC addresses, the 5th byte is 7:0 and the 6th byte is 15:8. The upper 10-bit are the bits 57:48 of the memory.	R/W

4.4.66 LK_LEARNCOUNT — Learned Address Count Register

Software would want to increment the value LEARNCOUNT when it has added a MAC address into the memory. It would want to decrement the value if an entry has been removed.

The increment/decrement bits (write_mode) ensure atomic operations on the counter to avoid inconsistency that would otherwise occur when using a read modify write scheme as the hardware could change the value meanwhile.

Address: 4405 0418h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	write_mode		—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	LEARNCOUNT													
Value after reset	X	X	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.68 LK_LEARNCOUNT Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b30	write_mode	The bits define how the LEARNCOUNT value is modified when writing into the register: 00b: Sets LEARNCOUNT to given value. 01b: Increments the LEARNCOUNT value by 1 (atomic). 10b: Decrements the LEARNCOUNT value by 1 (atomic). 11b: Reserved, write has no effect on LEARNCOUNT.	R/W
b29 to b14	Reserved	Set to zero on Write. ignore on Read.	R
b13 to b0	LEARNCOUNT	The Number of Learned Addresses	R/W

4.4.67 LK_AGETIME — Period of the Aging Timer

The timer is decremented once every 1024 switch system clock cycles and reloaded with this value when it reaches 0. One table entry is inspected whenever the agetimer expires.

Hence a full aging timeout is the amount of time given in this register multiplied by the size of the lookup memory (number of entries).

Default:

16384 (for example, when using a 200 MHz switch system clock, it is up to 11.5 minutes for 8192 entries table: n × 1024 × period × tablesize = 16384 × 1024 × 5 ns × 8192)

Address: 4405 041Ch

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	LK_AGETIME							
Value after reset	X	X	X	X	X	X	X	X	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	LK_AGETIME															
Value after reset	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.69 LK_AGETIME Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b24	Reserved	Set to zero on Write. ignore on Read.	R
b23 to b0	LK_AGETIME	24-bit Timer Value.	R/W

4.4.68 MGMT_TAG_CONFIG — Management Tag Configuration Register

Address: 4405 0480h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	Tagfield															
Value after reset	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	enable_type2	enable_type1	—	—	all_frames	enable
Value after reset	X	X	X	X	X	X	X	X	X	X	0	0	X	X	0	0

Table 4.70 MGMT_TAG_CONFIG Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b16	Tagfield	The value of the tag that is found in the first type/length field of the frame to identify that the control information is present within a frame. (31:24 = first octet, 23:16 = 2nd octet)	R/W
b15 to b6	Reserved	Set to zero on Write. ignore on Read.	R
b5	enable_type2	When set, frames with a type field that match the value in register PRIORITY_TYPE2[15:0] will get the control tag inserted. This is in addition to BPDU frames (which will have tag inserted always). ***Note)*** The valid bit (16) in PRIORITY_TYPE2 has no effect for this Type check. Validity is indicated when this enable Type2 bit is set.	R/W
b4	enable_type1	When set, frames with a type field that match the value in register PRIORITY_TYPE1[15:0] will get the control tag inserted. This is in addition to BPDU frames (which will have tag inserted always). ***Note)*** The valid bit (16) in PRIORITY_TYPE1 has no effect for this Type check. Validity is indicated when this enable Type1 bit is set. See also PRIORITY_TYPE register*1 description.	R/W
b3, b2	Reserved	Set to zero on Write. ignore on Read.	R
b1	all_frames	Enable Tag Insertion for All Frames If set, the function inserts the control tag into all transmitted frames at the management port. This requires that the CPU must be capable of receiving frames of at least 1526 bytes (1518 + 8) for normal frames and 1530 bytes for VLAN tagged frames. If 0 (reset default), only frames marked as management frames (BPDU frames) will be tagged with extra control information. Other frames are not manipulated. ***Note)*** The input processing (receive; CPU to A5PSW) operates always when enabled (bit0 = 1), independent from this setting.	R/W
b0	enable	Enable Management Port Tag Insertion Module If set, the function is adding tags to outgoing frames and removes tags from incoming frames at the management port. If 0 (reset default), no frame manipulation occurs. ***Note)*** When enabled, frame manipulation requires a recalculation of the CRC. Hence all MACs must be configured to remove CRC from incoming frames and add CRC on all outgoing frames.	R/W

Note 1. See **Section 4.4.45, PRIORITY_TYPE1 — Priority Type Register 1** and **Section 4.4.46, PRIORITY_TYPE2 — Priority Type Register 2**.

4.4.69 PEERDELAY[n] — Peer Delay Value for Port [n] (n = 0..3)

NOTE

This register is the value for line port[n]. It must be set by the application after measuring the latency at the port where it receives SYNC messages.

Address: 4405 050Ch + 4h × n

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	PEERDELAY[n]													
Value after reset	X	X	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PEERDELAY[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.71 PEERDELAY[n] Register Contents

Bit Position	Bit Name	Function	R/W
b31, b30	Reserved	Set to zero on Write. ignore on Read.	R
b29 to b0	PEERDELAY[n]	Peer Delay Value Determined at the Port The value is added to the received frame's correction field when a transient time correction is done. This time value is set by the application after performing peer delay measurement on the port. Required only when the network uses peer to peer transparent clocks. A value of 0 is necessary to implement an end to end transparent clock. Value is in nanoseconds and must be less than 10^9.	R/W

4.4.70 PORT[n]_CTRL — PORT[n] Timestamp Control/Status (n = 0..3)

Address: 4405 0520h + 8h × n

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	TS_KEEP	TS_OVR	TS_VALID
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	0	0	0

Table 4.72 PORT[n]_CTRL Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b3	Reserved	Set to zero on Write. ignore on Read.	R
b2	TS_KEEP	Configuration bit: Keep last timestamp in the transmit timestamp registers. When cleared (default), a new timestamp overwrites a stored value. When this bit is set, the first timestamp stored is kept and following timestamps are ignored until software has processed the timestamp (i.e. wrote this register to clear the valid bit).	R/W
b1	TS_OVR	A newer timestamp has overwritten the last stored timestamp. This occurs, if a valid timestamp is stored (TS_VALID = 1) and another timestamp is received before the software could read the value and clear the valid bit. Writing to the register (with any value) clears the bit. When the configuration bit TS_KEEP is set, this bit indicates that timestamp(s) were received but ignored (i.e. a new timestamp was received while TS_VALID = 1).	R/W
b0	TS_VALID	A valid timestamp is available. Writing to the register (with any value) clears the bit.	R/W

4.4.71 PORT[n]_TIME — PORT[n] Memorized Transmit Timestamp (n = 0..3)

Address: 4405 0524h +8h × n

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	PORT[n]_TIME															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PORT[n]_TIME															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.73 PORT[n]_TIME Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	PORT[n]_TIME	PORT[n] Memorized Transmit Timestamp	R

4.4.72 INT_CONFIG — Interrupt Enable Configuration Register

Address: 4405 0600h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	PATTERN_INT	TDMA_INT	—	—	IRQ_MAC_EEE				—	—	—	—	IRQ_TSM_TX			
Value after reset	0	0	X	X	0	0	0	0	X	X	X	X	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	IRQ_LINK				HUB_INT	PRP_INT	DLR_INT	IRQ_TEST	LK_NEW_SRC	—	MDIO1	IRQ_EN
Value after reset	X	X	X	X	0	0	0	0	0	0	0	0	0	X	0	0

Table 4.74 INT_CONFIG Register Contents (1/2)

Bit Position	Bit Name	Function	R/W
b31	PATTERN_INT	When set, the RX Pattern Matcher module's interrupt output is wired (ORed) with the host interrupt signal (A5PSW_Int). When set and any of the pattern interrupts asserts, the host interrupt will also assert. No latching occurs and writing the bit 31 of the INT_STAT_ACK register has no effect. To clear the interrupt, the PTN_IRQ_STAT_ACK register must be written instead. ***Note)*** Does not influence the function of the A5PSW_PTRN_Int signal.	R/W
b30	TDMA_INT	When set, the TDMA scheduler interrupt is wired (ORed) with the host interrupt signal (A5PSW_Int). When set and any of the TDMA interrupts asserts, the host interrupt will assert. No latching occurs and writing the bit 30 of the global INT_STAT_ACK register has no effect. To clear the interrupt, the TDMA_IRQ_STAT_ACK register must be written instead.	R/W
b29, b28	Reserved	Set to zero on Write. ignore on Read.	R
b27 to b24	IRQ_MAC_EEE	Per Line Port MAC interrupt When set, an interrupt is generated when the port's EEE function changed state. Bit 24 = Port 0, Bit 25 = Port 1, Bit 26 = Port 2, Bit 27 = Port 3.	R/W
b23 to b20	Reserved	Set to zero on Write. ignore on Read.	R
b19 to b16	IRQ_TSM_TX	Per Line Port Transmit Timestamp Capture Interrupt Enable When set, an interrupt is generated when the port's transmit timestamp register has stored a new timestamp (see **Section 4.5.4, Timestamping Functions (TSM)**). One bit for each line port: Bit 16 = Port 0, Bit 17 = Port 1, Bit 18 = Port 2, Bit 19 = Port 3.	R/W
b15 to b12	Reserved	Set to zero on Write. ignore on Read.	R
b11 to b8	IRQ_LINK	Per Line Port Phy Link Change Interrupt Enable When set, an interrupt is generated when the Link status input from PHY of a port changes. One bit for each line port: Bit 8 = Port 0, Bit 9 = Port 1, Bit 10 = Port 2, Bit 11 = Port 3.	R/W
b7	HUB_INT	When set, the Hub module's interrupt output (A5PSW_HUB_Int) is wired (ORed) with the host interrupt signal (A5PSW_Int). When set and A5PSW_HUB_Int asserts, the host interrupt will also assert. No latching occurs and writing the bit 7 of the INT_STAT_ACK register has no effect. To clear the interrupt, the HUB_IRQ_STAT_ACK register must be written instead. ***Note)*** Does not influence the function of the A5PSW_HUB_Int signal.	R/W
b6	PRP_INT	When set, the PRP module's interrupt output (A5PSW_PRP_Int) is wired (ORed) with the host interrupt signal (A5PSW_Int). When set and A5PSW_PRP_Int asserts, the host interrupt will also assert. No latching occurs and writing the bit 6 of this register has no effect. To clear the interrupt, the PRP_IRQ_STAT_ACK register must be written instead. ***Note)*** This bit does not influence the function of the A5PSW_PRP_Int signal.	R/W

Table 4.74 INT_CONFIG Register Contents (2/2)

Bit Position	Bit Name	Function	R/W
b5	DLR_INT	When set, the DLR module's interrupt output (A5PSW_DLR_Int) is wired (ORed) with the host interrupt signal (A5PSW_Int). When set and A5PSW_DLR_Int asserts, the host interrupt will also assert. No latching occurs and writing the bit 5 of the INT_STAT_ACK register has no effect. To clear the interrupt, the DLR_IRQ_STAT_ACK register must be written instead. This may be used if all interrupt handling should be done through the single interrupt signal (A5PSW_Int) and the separate A5PSW_DLR_Int signal is not used. (see also DLR_IRQ_CONTROL and DLR_IRQ_STAT_ACK). ***Note*)** This bit does not influence the function of the A5PSW_DLR_Int signal.	R/W
b4	IRQ_TEST	When set, an interrupt is triggered immediately. Can be used to cause a software controlled interrupt for testing purposes.	R/W
b3	LK_NEW_SRC	Enable Interrupt for New Source Address If this bit is set, an interrupt is triggered when the lookup process detects a new (unknown) source address on a port. ***Note*)** The interrupt triggers on every unknown source detected, independent of learning (INPUT_LEARN_BLOCK) or source discard (LK_CTRL) was enabled or not.	R/W
b2	Reserved	Set to zero on Write. ignore on Read.	R
b1	MDIO1	Enable Interrupt on Transaction Complete from MDIO Controller	R/W
b0	IRQ_EN	Interrupt Global Enable When set and if any of the interrupt events occurs and is enabled, the interrupt signal is asserted (1).	R/W

4.4.73 INT_STAT_ACK — Interrupt Status/ACK Register

The register provides the current status of the interrupts and allows clearing/acknowledging one or all interrupts. The interrupt signal is asserted only if the corresponding interrupt has been enabled within the INT_CONFIG register. Each interrupt latch bit in this register asserts when the interrupt event occurred, independent of the enable setting within INT_CONFIG.

Address: 4405 0604h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	PATTERN_INT	TDMA_INT	interrupt_latches													
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	interrupt_latches								HUB_INT	PRP_INT	DLR_INT	IRQ_TEST	LK_NEW_SRC	—	MDIO1	IRQ_PEND
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.75 INT_STAT_ACK Register Contents (1/2)

Bit Position	Bit Name	Function	R/W
b31	PATTERN_INT	Interrupt Pending Status from RX Pattern Matcher Module Asserted as long as the module's interrupt register has pending interrupts (represents OR of all RX Pattern Matcher module's interrupt output). To clear the interrupt, the pattern module's status/acknowledge register must be written. Writing this bit has no effect. See PTN_IRQ_STAT_ACK.	R
b30	TDMA_INT	Interrupt Pending Status from TDMA Scheduler Asserted as long as any TDMA interrupt has been enabled and is pending. To clear the interrupt, the TDMA_IRQ_STAT_ACK register must be used. Writing this bit has no effect.	R
b29 to b8	interrupt_latches	Latched interrupt status for each bit as defined in register INT_CONFIG[29:8]. Writing a bit with 1 clears the interrupt.	R/W
b7	HUB_INT	Interrupt Pending Status from Hub Module Asserted as long as the Hub module's interrupt register has pending interrupts (represents the status of signal A5PSW_HUB_Int). To clear the interrupt, the Hub module's status/acknowledge register must be written. Writing this bit has no effect. See HUB_IRQ_STAT_ACK.	R
b6	PRP_INT	Interrupt Pending Status from PRP Module Asserted as long as the PRP module's interrupt register has pending interrupts (represents the status of signal A5PSW_PRP_Int). To clear the interrupt, the PRP module's status/acknowledge register must be written. Writing this bit has no effect.	R
b5	DLR_INT	Interrupt Pending Status from DLR Module Asserted as long as the DLR module's interrupt register has pending interrupts (represents the status of signal A5PSW_DLR_Int). To clear the interrupt, the DLR module's status/acknowledge register must be written. Writing this bit has no effect.	R
b4	IRQ_TEST	Interrupt Status for IRQ_TEST Is set as long as INT_CONFIG.IRQ_TEST is 1. To clear the interrupt, the INT_CONFIG.IRQ_TEST. must be cleared.	R
b3	LK_NEW_SRC	Latched Interrupt Status for LK_NEW_SRC Writing the bit with 1 clears the latch.	R/W
b2	Reserved	Set to zero on Write. ignore on Read.	R
b1	MDIO1	Latched Interrupt Status for MDIO1 Writing the bit with 1 clears the latch.	R/W

Table 4.75 INT_STAT_ACK Register Contents (2/2)

Bit Position	Bit Name	Function	R/W
b0	IRQ_PEND	Interrupt Pending Status Asserted as long as any pending and enabled interrupt exists. Each interrupt latch bit in this register asserts when the interrupt event occurred, independent of the enable setting within INT_CONFIG. However the IRQ_PEND asserts only if a latch bit and its corresponding enable bit are both set for any of the events. ***Note)*** When set, the signal (A5PSW_Int) will assert if the global enable bit INT_CONFIG[0] is set.	R

4.4.74 MDIO_CFG_STATUS — MDIO Configuration and Status Register

Address: 4405 0700h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CLKDIV									—	DISPREAM	HOLD			READERR	BUSY
Value after reset	0	0	0	1	0	1	0	0	0	X	0	0	0	0	0	0

Table 4.76 MDIO_CFG_STATUS Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b16	Reserved	Set to zero on Write. ignore on Read.	R
b15 to b7	CLKDIV	MDIO Clock Divisor A value of 5 to 511 can be set. The frequency is (AHB clock freq.)/((2 × divisor)+1). The reset default is 40. Setting the divisor to 0 disables MDC.	R/W
b6	Reserved	Set to zero on Write. ignore on Read.	R
b5	DISPREAM	0: Enable Preamble 1: Disable Preamble	R/W
b4 to b2	HOLD	MDIO Hold Time Setting 000b: 1 AHB clock cycle (default) 001b: 3 AHB clock cycles 010b: 5 AHB clock cycles 011b: 7 AHB clock cycles 100b: 9 AHB clock cycles 101b: 11 AHB clock cycles 110b: 13 AHB clock cycles 111b: 15 AHB clock cycles	R/W
b1	READERR	MDIO Read Error (Read Only Bit) If set, the last read transaction had no response from a PHY and the data read could be invalid. This can happen, if the PHY address does not match any PHY that is available on the MDIO bus.	R
b0	BUSY	MDIO Busy (Read Only Bit) If set, a MDIO transaction is currently ongoing. If cleared, the application can access the other registers.	R

4.4.75 MDIO_COMMAND — MDIO PHY Command Register

Address: 4405 0704h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TRANINIT	—	—	—	—	—	PHYADDR					REGADDR				
Value after reset	0	X	X	X	X	X	0	0	0	0	0	0	0	0	0	0

Table 4.77 MDIO_COMMAND Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b16	Reserved	Set to zero on Write. ignore on Read.	R
b15	TRANINIT	If written with 1, a read transaction is initiated.	R/W
b14 to b10	Reserved	Set to zero on Write. ignore on Read.	R
b9 to b5	PHYADDR	PHY Address	R/W
b4 to b0	REGADDR	Register Address	R/W

4.4.76 MDIO_DATA — MDIO Data Register

Address: 4405 0708h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MDIO_DATA															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.78 MDIO_DATA Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b16	Reserved	Set to zero on Write. ignore on Read.	R
b15 to b0	MDIO_DATA	• When written: Initiates a write transaction to the PHY. The MDIO_COMMAND register must have been initialized. The busy status bit will be set immediately and cleared when the write transaction has finished. • When read: Returns the data read from the PHY register after a read transaction has been completed (initiated by writing the MDIO_COMMAND register).	R/W

4.4.77 REV_P[n] — PORT[n] MAC Core Revision (n = 0..4)

Address: 4405 0800h + 400h × n

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	REV															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	REV															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Table 4.79 REV_P[n] Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	REV	Revision	R

4.4.78 COMMAND_CONFIG_P[n] — Port[n] Command Configuration Register (n = 0..4)

Address: 4405 0808h + 400h × n

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	ENA_10	NO_LGTH_CHECK	CNTL_FRM_ENA	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	0	1	0	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	LOOP_ENA	—	SW_RESET	—	TX_CRC_APPEND	HD_ENA	TX_ADDR_INS	PAUSE_IGNORE	PAUSE_FWD	CRC_FWD	PAD_EN	PROMIS_EN	ETH_SPEED	—	RX_ENA	TX_ENA
Value after reset	0	0	0	X	1	0	0	0	0	0	0	1	0	X	1	1

Table 4.80 COMMAND_CONFIG_P[n] Register Contents (1/2)

Bit Position	Bit Name	Function	R/W
b31 to b26	Reserved	Set to zero on Write. ignore on Read.	R
b25	ENA_10	This bit has no effect except PHYSPEED bit of STATUS_P[n] register. When using the HUB function, set these bits as they are reflected appropriately in STATUS_P[n].PHYSPEED. See STATUS_P[n].PHYSPEED for speed settings.	R/W
b24	NO_LGTH_CHECK	Payload Length Check Disable When set to “0”, the Core checks the frame’s payload length with the Frame Length/Type field. When set to “1”, the payload length check is disabled.	R/W
b23	CNTL_FRM_ENA	MAC Control Frame Enable When set to “0”, MAC Control frames (type = 0x8808) with any opcode other than 0x0001 (Pause Frame) are silently discarded. When set to “1”, MAC Control frames with any opcode other than 0x0001 are accepted and forwarded to the Client interface.	R/W
b22 to b16	Reserved	Set to zero on Write. ignore on Read.	R
b15	LOOP_ENA	Set to zero on Write. ignore on Read.	R/W
b14	Reserved	Set to zero on Write. ignore on Read.	R
b13	SW_RESET	Self Clearing Reset Command Bit When written with 1, it clears the statistics registers and disables the MAC transmit and receive datapath. ***Note)*** Only if this bit is written in the MAC Port0 register, it will clear all statistics of all ports. Writing this bit on any other port's register has no effect on the statistics counters.	R/W
b12	Reserved	Set to zero on Write. ignore on Read.	R
b11	TX_CRC_APPEND	Enable CRC Append on Transmit If set (1), the TX will append a CRC to all outgoing frames. If frame manipulation operations are activated within the switch, all MACs should be configured to remove the CRC on receive (bit CRC_FWD = 0) and append CRC on transmit. If no frame manipulation is configured within the switch, the CRC may be forwarded together with the frame throughout the switch (i.e. all MACs can be configured with CRC_FWD = 1 and TX_CRC_APPEND = 0)	R/W
b10	HD_ENA	Enable auto full/half-duplex operation (1) or full-duplex only (0). To set half-duplex, this bit is set to 1 and PHY_DUPLEX bit of SWDUPC register in Ethernet Accessory Register is set to 0. Otherwise, full-duplex is selected.	R/W
b9	TX_ADDR_INS	Non writeable bit, Fixed to 0 always.	R
b8	PAUSE_IGNORE	Ignore Pause Frame Quanta If enabled (Set to “1”), received pause frames are ignored by the MAC. When disabled (Set to reset value “0”), the transmit process is stopped for the amount of time specified in the pause quanta received within the pause frame.	R/W

Table 4.80 COMMAND_CONFIG_P[n] Register Contents (2/2)

Bit Position	Bit Name	Function	R/W
b7	PAUSE_FWD	Terminate/Forward Pause Frames If enabled (Set to “1”), pause frames are forwarded to the user application. In normal mode (Set to reset value “0”), pause frames are terminated and discarded in the MAC. This bit is fixed to 0 always.	R
b6	CRC_FWD	Terminate/Forward Received CRC If enabled (1), the CRC field of received frames is forwarded with the frame to the user application. If disabled (Set to reset value “0”), the CRC field is stripped from the frame.	R/W
b5	PAD_EN	Enable/Disable Frame Padding Remove on Receive If enabled (1), padding is removed from received frames before they are given to the user application. If disabled (set to reset value “0”), no padding is removed on receive by the MAC. This bit is fixed to 0 always.	R
b4	PROMIS_EN	Enable/Disable MAC Promiscuous Operation When asserted (Set to “1”), all frames are received without Unicast address filtering. This bit is fixed to 1 always.	R
b3	ETH_SPEED	Defines 10/100 Mode (0) or Gigabit Mode (1) of operation.	R/W
b2	Reserved	Set to zero on Write. ignore on Read.	R
b1	RX_ENA	Enable/Disable MAC Receive Path When set to “0”, the MAC receive function is disabled. When set to “1”, the MAC receive function is enabled.	R/W
b0	TX_ENA	Enable/Disable MAC Transmit Path When set to “0”, the MAC transmit function is disabled. When set to “1”, the MAC transmit function is enabled.	R/W

4.4.79 MAC_ADDR_0_P[n] — Port[n] MAC Address Register 0 (n = 0..3)

Address: 4405 080Ch + 400h × n

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	MAC_ADDR															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MAC_ADDR															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.81 MAC_ADDR_0_P[n] Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	MAC_ADDR	The first 4 bytes of the port's MAC address. First byte is (7:0). The MAC address is used on locally generated frames (e.g. pause frames, peer-delay response).	R/W

4.4.80 MAC_ADDR_1_P[n] — Port[n] MAC Address Register 1 (n = 0..3)

Address: 4405 0810h + 400h × n

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MAC_ADDR															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.82 MAC_ADDR_1_P[n] Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b16	Reserved	Set to zero on Write. ignore on Read.	R
b15 to b0	MAC_ADDR	The last 2 bytes of the port's MAC address. (7:0) is 5th byte and (15:8) is 6th byte.	R/W

4.4.81 FRM_LENGTH_P[n] — PORT[n] Maximum Frame Length Register (n = 0..4)

NOTE

To allow the management port to accept a frame with the special management tag, the allowed length must be increased by 8 for this port.

Address: 4405 0814h + 400h × n

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	FRM_LENGTH													
Value after reset	X	X	0	0	0	1	0	1	1	1	1	1	1	1	1	0

Table 4.83 FRM_LENGTH_P[n] Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b14	Reserved	Set to zero on Write. ignore on Read.	R
b13 to b0	FRM_LENGTH	Maximum Frame Length (RW) Defines a 14 Bit maximum frame length used by the MAC receive logic to check frames. This is the maximum total length of a frame allowed. Hence, if VLAN frames should be supported, the value must be adjusted to 1522 or 1526 for single or double tagged VLAN support respectively.	R/W

4.4.82 PAUSE_QUANT_P[n] — PORT[n] MAC Pause Quanta (n = 0..4)

Address: 4405 0818h + 400h × n

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PAUSE_QUANT															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Table 4.84 PAUSE_QUANT_P[n] Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b16	Reserved	Set to zero on Write. ignore on Read.	R
b15 to b0	PAUSE_QUANT	Pause Quanta 16 Bit value, sets, in increment of 512 Ethernet bit times, the pause quanta used in each Pause Frame sent to the remote Ethernet device.	R

4.4.83 PTPClockIdentity1_P[n] — PORT[n] PTP Clock Identity1 Register (n = 0..3)

Address: 4405 0830h + 400h × n

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	ClockIdentity3								ClockIdentity2							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ClockIdentity1								ClockIdentity0							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.85 PTPClockIdentity1_P[n] Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b24	ClockIdentity3	23, portIdentity.ClockIdentity[3]	R/W
b23 to b16	ClockIdentity2	22, portIdentity.ClockIdentity[2]	R/W
b15 to b8	ClockIdentity1	21, portIdentity.ClockIdentity[1]	R/W
b7 to b0	ClockIdentity0	20, portIdentity.ClockIdentity[0]	R/W

4.4.84 PTPClockIdentity2_P[n] — PORT[n] PTP Clock Identity2 Register (n = 0..3)

Address: 4405 0834h + 400h × n

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	ClockIdentity7								ClockIdentity6							
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ClockIdentity5								ClockIdentity4							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.86 PTPClockIdentity2_P[n] Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b24	ClockIdentity7	27, portIdentity.ClockIdentity[7]	R/W
b23 to b16	ClockIdentity6	26, portIdentity.ClockIdentity[6]	R/W
b15 to b8	ClockIdentity5	25, portIdentity.ClockIdentity[5]	R/W
b7 to b0	ClockIdentity4	24, portIdentity.ClockIdentity[4]	R/W

4.4.85 PTPAutoResponse_P[n] — PORT[n] PTP Auto Response Register (n = 0..3)

Address: 4405 0838h + 400h × n

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	PortNumber1								PortNumber0							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	AutoRespEnable
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	0

Table 4.87 PTPAutoResponse_P[n] Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b24	PortNumber1	28, portIdentity.PortNumber[0] (msb)	R/W
b23 to b16	PortNumber0	29, portIdentity.PortNumber[1] (lsb)	R/W
b15 to b1	Reserved	Set to zero on Write. ignore on Read.	R
b0	AutoRespEnable	Enable automatic generation of IEEE 1588v2 Layer2 peer delay response (PDELAY_RESP) messages. When 1, the MAC detects PDELAY_REQ messages causing an automatic reply with PDELAY_RESP. When 0, the MAC will not cause automatic response generation. ***Note)*** The PRIORITY_TYPE registers can be used to define the priority for the response frames.	R/W

4.4.86 STATUS_P[n] — PORT[n] Port Status Register (n = 0..4)

Address: 4405 0840h + 400h × n

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	PHYDUPLEX	PHYLINK	PHYSPEED	
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	1	0	X	X

Table 4.88 STATUS_P[n] Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b4	Reserved	Set to zero on Write. ignore on Read.	R
b3	PHYDUPLEX	Duplex status from PHY interface (1 = full-duplex, 0 = half-duplex) The information is valid only if the PHYLINK = 1. Otherwise it can be arbitrary. ***Note)*** This is a direct representation of PHY_DUPLEX bits of SWDUPC register in Ethernet Accessory Register. Therefore, actual PHY status may differ from this.	R
b2	PHYLINK	Link status from PHY interface (1 = link up, 0 = link down) ***Note)*** This is a direct representation of Link status input from PHY.	R
b1, b0	PHYSPEED	Currently active PHY interface speed: 00b = 10 Mbps 01b = 100 Mbps 10b = 1 Gbps 11b = reserved, unused Define OR'ed of the configuration bits for each port as follows. (n: port) mode10 = (COMMAND_CONFIG_P[n].ENA_10 OR SWCTRL.SET10[n]) mode1000 = (COMMAND_CONFIG_P[n].ETH_SPEED OR SWCTRL.SET1000[n]) mode1000 = 1 and mode10 = x: 1 Gbps mode1000 = 0 and mode10 = 0: 100 Mbps mode1000 = 0 and mode10 = 1: 10 Mbps ***Note)*** The information is valid only if the link is up (PHYLINK = 1). Otherwise it can be arbitrary.	R

4.4.87 TX_IPG_LENGTH_P[n] — PORT[n] Transmit IPG Length Register (n = 0..4)

Address: 4405 0844h + 400h × n

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	TX_IPG_LENGTH				
Value after reset	X	X	X	X	X	X	X	X	X	X	X	0	1	1	0	0

Table 4.89 TX_IPG_LENGTH_P[n] Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b5	Reserved	Set to zero on Write. ignore on Read.	R
b4 to b0	TX_IPG_LENGTH	define transmit interpacket gap in octets. Allowed values are in the range of 8 to 31	R/W

4.4.88 EEE_CTL_STAT_P[n] — PORT[n] MAC EEE Functions Control and Status (n = 0..3)

Address: 4405 0848h + 400h × n

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	STLH_LPI_IND	—	STLH_TXBUSY	STLH_LPI_TXHOLD	STLH_LPI_REQ
Value after reset	X	X	X	X	X	X	X	X	X	X	X	0	X	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	ST_LPI_IND	ST_TXAVAIL	ST_TXBUSY	ST_LPI_TXHOLD	ST_LPI_REQ	—	—	—	—	—	LPI_TXHOLD	LPI_REQ	EEE_AUTO
Value after reset	X	X	X	0	0	0	0	0	X	X	X	X	X	0	0	0

Table 4.90 EEE_CTL_STAT_P[n] Register Contents (1/2)

Bit Position	Bit Name	Function	R/W
b31 to b21	Reserved	Set to zero on Write. ignore on Read.	R
b20	STLH_LPI_IND	Status (latched high) of Received LPI (ST_LPI_IND) Bit clears on register read.	R
b19	Reserved	Set to zero on Write. ignore on Read.	R
b18	STLH_TXBUSY	Status (latched high) if the MAC is/was Transmitting Bit clears on register read.	R
b17	STLH_LPI_TXHOLD	Status (latched high) of INTERNAL LPI_TXHOLD to the MAC Bit clears on register read.	R
b16	STLH_LPI_REQ	Status (latched high) of Internal LPI_REQ to the MAC Bit clears on register read.	R
b15 to b13	Reserved	Set to zero on Write. ignore on Read.	R
b12	ST_LPI_IND	Status (real time) of Received LPI Asserts as long as the RS layer of the MAC detects LPI sequences at it's receive interface.	R
b11	ST_TXAVAIL	Status (real time) if the MAC transmit FIFO has data available for transmission.	R
b10	ST_TXBUSY	Status (real time) if the MAC is currently transmitting.	R
b9	ST_LPI_TXHOLD	Status (real time) of Internal LPI_TXHOLD to the MAC Indicates the MAC is stopped and no longer serving frames. It will assert either as a result of writing the LPI_REQ or LPI_TXHOLD bit, or when the automatic mode is enabled and the MAC is not yet allowed to transmit frames.	R
b8	ST_LPI_REQ	Status (real time) of Internal LPI_REQ to the MAC It will assert either as a result of writing the LPI_REQ bit, or when the automatic mode is enabled and the idle time has expired.	R
b7 to b3	Reserved	Set to zero on Write. ignore on Read.	R
b2	LPI_TXHOLD	MAC Transmission Hold When 1, the MAC will continue transmitting a currently ongoing frame, if any, and then stop reading any more frames from its transmit FIFO/Queue. This bit must be set before LPI_REQ is cleared to ensure the MAC transitions to idle after LPI. Then LPI_TXHOLD is cleared after the (application controlled) wake time expired, allowing the MAC to transmit frames normally again. ***Note)*** Using LPI_REQ and LPI_TXHOLD controls requires the application to implement corresponding quiet and wake timers. Set 0 for normal operation or when EEE_AUTO is used.	R/W

Table 4.90 EEE_CTL_STAT_P[n] Register Contents (2/2)

Bit Position	Bit Name	Function	R/W
b1	LPI_REQ	Request LPI Transmission When MAC Becomes Idle When 1, the MAC will continue transmitting any available frames in its output Queue(s) and then transmit LPI sequences. When it transitioned into LPI, the ST_LPI_REQ status bit asserts. As long as in LPI state, the MAC will not serve new frames arriving in its output queue(s). Writing LPI_REQ = 1 and LPI_TXHOLD = 0 will ensure all frames in the transmit queue are sent before entering LPI. If LPI_TXHOLD is set together with LPI_REQ, the MAC will stop immediately after currently ongoing frame transmission completed and not empty its output queue. Before clearing LPI_REQ, the LPI_TXHOLD bit should be set to prevent the MAC from immediately transmitting frames following LPI removal (i.e. not respecting any wake time). Hence a typical application controlled power down sequence is: LPI_REQ = 1 -> wait until ST_LPI_REQ = 1 -> LPI_TXHOLD = 1 -> wait quiet timeout or data available -> LPI_REQ = 0 -> wait wake timeout -> LPI_TXHOLD = 0. This bit can be set at any time (even when EEE_AUTO = 1). Set 0 for normal operation.	R/W
b0	EEE_AUTO	EEE automatic mode of operation. When 1, the MAC will begin transmitting LPI sequences as soon as it has been in idle for the duration configured in register EEE_IDLE_TIME_P[n]. The bit should be set 1 only following initialization of the timer registers (EEE_IDLE_TIME_P[n], EEE_TWSYS_TIME_P[n]). When 0, the MAC will not perform automatic LPI transmission.	R/W

4.4.89 EEE_IDLE_TIME_P[n] — PORT[n] EEE Idle Time Register (n = 0..3)

Address: 4405 084Ch + 400h × n

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	EEE_IDLE_TIME_P[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	EEE_IDLE_TIME_P[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.91 EEE_IDLE_TIME_P[n] Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	EEE_IDLE_TIME_P[n]	Time(−1) the transmitter must be idle before transmission of LPI will begin. 32-bit value in steps of 32 switch system clock cycles. A value of 0 disables the timer. The value must be set to less 1 count.	R/W

4.4.90 EEE_TWSYS_TIME_P[n] — PORT[n] EEE Wake Up Time Register (n = 0..3)

Address: 4405 0850h + 400h × n

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	EEE_TWSYS_TIME_P[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	EEE_TWSYS_TIME_P[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.92 EEE_TWSYS_TIME_P[n] Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	EEE_TWSYS_TIME_P[n]	Time(−1) after PHY wakeup until the MAC is allowed to begin transmitting the first frame again. 32-bit value in steps of switch system clock cycles. A value of 0 disables the timer. The value must be set to less 1 count.	R/W

4.4.91 IDLE_SLOPE_P[n] — PORT[n] MAC Traffic Shaper Bandwidth Control (n = 0..4)

Address: 4405 0854h + 400h × n

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	IDLE_SLOPE										
Value after reset	X	X	X	X	X	0	0	0	0	0	0	0	0	0	0	0

Table 4.93 IDLE_SLOPE_P[n] Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b11	Reserved	Set to zero on Write. ignore on Read.	R
b10 to b0	IDLE_SLOPE	Traffic Shaper Bandwidth Control When a non zero value is set, the shaper is active and controls the bandwidth. When 0, the shaper is disabled. Valid values are 2..2046.	R/W

4.4.92 aFramesTransmittedOK_P[n] — PORT[n] MAC Transmitted Valid Frame Count Register (n = 0..4)

Address: 4405 0868h + 400h × n

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	aFramesTransmittedOK_P[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	aFramesTransmittedOK_P[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.94 aFramesTransmittedOK_P[n] Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	aFramesTransmitted OK_P[n]	PORT[n], this field indicates the number of MAC Valid Transmitted (incl. pause)	R

4.4.93 aFramesReceivedOK_P[n] —PORT[n] MAC Received Valid Frame Count Register (n = 0..4)

Address: 4405 086Ch + 400h × n

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	aFramesReceivedOK_P[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	aFramesReceivedOK_P[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.95 aFramesReceivedOK_P[n] Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	aFramesReceivedOK_P[n]	PORT[n], this field indicates the number of MAC Valid Received (incl. pause)	R

4.4.94 aFrameCheckSequenceErrors_P[n] —PORT[n] MAC FCS Error Frame Count Register (n = 0..4)

Address: 4405 0870h + 400h × n

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	aFrameCheckSequenceErrors_P[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	aFrameCheckSequenceErrors_P[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.96 aFrameCheckSequenceErrors_P[n] Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	aFrameCheckSequenceErrors_P[n]	PORT[n], this field indicates the number of MAC Valid Length but CRC Error	R

4.4.95 aAlignmentErrors_P[n] —PORT[n] MAC Alignment Error Frame Count Register (n = 0..4)

Address: 4405 0874h + 400h × n

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	aAlignmentErrors_P[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	aAlignmentErrors_P[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.97 aAlignmentErrors_P[n] Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	aAlignmentErrors_P[n]	PORT[n], this field indicates the number of MAC Odd Number of Nibbles (MII) Received	R

4.4.96 aOctetsTransmittedOK_P[n] —PORT[n] MAC Transmitted Valid Frame Octets Register (n = 0..4)

Address: 4405 0878h + 400h × n

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	aOctetsTransmittedOK_P[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	aOctetsTransmittedOK_P[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.98 aOctetsTransmittedOK_P[n] Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	aOctetsTransmittedOK_P[n]	PORT[n], this field indicates the octets of MAC valid transmitted.	R

4.4.97 aOctetsReceivedOK_P[n] —PORT[n] MAC Received Valid Frame Octets Register (n = 0..4)

Address: 4405 087Ch + 400h × n

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	aOctetsReceivedOK_P[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	aOctetsReceivedOK_P[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.99 aOctetsReceivedOK_P[n] Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	aOctetsReceivedOK_P[n]	PORT[n], this field indicates the octets of MAC Valid Received	R

4.4.98 aTxPAUSEMACCtrlFrames_P[n] —PORT[n] MAC Transmitted Pause Frame Count Register (n = 0..4)

Address: 4405 0880h + 400h × n

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	aTxPAUSEMACCtrlFrames_P[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	aTxPAUSEMACCtrlFrames_P[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.100 aTxPAUSEMACCtrlFrames_P[n] Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	aTxPAUSEMACCtrlFrames_P[n]	PORT[n], this field indicates the number of MAC valid pause transmitted.	R

4.4.99 aRxPAUSEMACCtrlFrames_P[n] — PORT[n] MAC Received Pause Frame Count Register (n = 0..4)

Address: 4405 0884h + 400h × n

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	aRxPAUSEMACCtrlFrames_P[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	aRxPAUSEMACCtrlFrames_P[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.101 aRxPAUSEMACCtrlFrames_P[n] Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	aRxPAUSEMACCtrlFrames_P[n]	PORT[n], this field indicates the number of MAC Valid Pause Received	R

4.4.100 ifInErrors_P[n] — PORT[n] MAC Input Error Count Register (n = 0..4)

Address: 4405 0888h + 400h × n

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	ifInErrors_P[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ifInErrors_P[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.102 ifInErrors_P[n] Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	ifInErrors_P[n]	PORT[n], this field indicates the number of MAC Any Error During Reception: CRC, Length, PHY Error, rx fifo Overflow	R

4.4.101 ifOutErrors_P[n] — PORT[n] MAC Output Error Count Register (n = 0..4)

Increments on internal errors (e.g. TX FIFO underflow) or when cut through forwarding was used and a frame with error was received.

NOTE

Does not count aborted frames (half-duplex collisions).

Address: 4405 088Ch + 400h × n

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	ifOutErrors_P[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ifOutErrors_P[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.103 ifOutErrors_P[n] Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	ifOutErrors_P[n]	PORT[n], this field indicates the number of MAC Frame Transmitted with PHY Error	R

4.4.102 ifInUcastPkts_P[n] — PORT[n] MAC Received Unicast Frame Count Register (n = 0..4)

Address: 4405 0890h + 400h × n

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	ifInUcastPkts_P[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ifInUcastPkts_P[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.104 ifInUcastPkts_P[n] Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	ifInUcastPkts_P[n]	PORT[n], this field indicates the number of MAC Unicast Frame Valid Received	R

4.4.103 ifInMulticastPkts_P[n] — PORT[n] MAC Received Multicast Frame Count Register (n = 0..4)

Address: 4405 0894h + 400h × n

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	ifInMulticastPkts_P[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ifInMulticastPkts_P[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.105 ifInMulticastPkts_P[n] Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	ifInMulticastPkts_P[n]	PORT[n], this field indicates the number of MAC Multicast Frame Valid Received	R

4.4.104 ifInBroadcastPkts_P[n] — PORT[n] MAC Received Broadcast Frame Count Register (n = 0..4)

Address: 4405 0898h + 400h × n

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	ifInBroadcastPkts_P[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ifInBroadcastPkts_P[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.106 ifInBroadcastPkts_P[n] Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	ifInBroadcastPkts_P[n]	PORT[n], this field indicates the number of MAC Broadcast Frame Valid Received	R

4.4.105 ifOutDiscards_P[n] — PORT[n] MAC Discarded Outbound Frame Count Register (n = 0..4)

Address: 4405 089Ch + 400h × n

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	ifOutDiscards_P[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ifOutDiscards_P[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.107 ifOutDiscards_P[n] Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	ifOutDiscards_P[n]	Not Applicable	R

4.4.106 ifOutUcastPkts_P[n] — PORT[n] MAC Transmitted Unicast Frame Count Register (n = 0..4)

Address: 4405 08A0h + 400h × n

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	ifOutUcastPkts_P[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ifOutUcastPkts_P[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.108 ifOutUcastPkts_P[n] Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	ifOutUcastPkts_P[n]	PORT[n], this field indicates the number of MAC Unicast Frame Valid Transmitted	R

4.4.107 ifOutMulticastPkts_P[n] — PORT[n] MAC Transmitted Multicast Frame Count Register (n = 0..4)

Address: 4405 08A4h + 400h × n

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	ifOutMulticastPkts_P[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ifOutMulticastPkts_P[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.109 ifOutMulticastPkts_P[n] Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	ifOutMulticastPkts_P[n]	PORT[n], this field indicates the number of MAC Multicast Frame Valid Transmitted	R

4.4.108 ifOutBroadcastPkts_P[n] — PORT[n] MAC Transmitted Broadcast Frame Count Register (n = 0..4)

Address: 4405 08A8h + 400h × n

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	ifOutBroadcastPkts_P[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ifOutBroadcastPkts_P[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.110 ifOutBroadcastPkts_P[n] Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	ifOutBroadcastPkts_P[n]	PORT[n], this field indicates the number of MAC Broadcast Frame Valid Transmitted	R

4.4.109 etherStatsDropEvents_P[n] — PORT[n] MAC Dropped Frame Count Register (n = 0..4)

Address: 4405 08ACh + 400h × n

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	etherStatsDropEvents_P[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	etherStatsDropEvents_P[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.111 etherStatsDropEvents_P[n] Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	etherStatsDropEvents_P[n]	PORT[n], this field indicates the number of MAC Rx FIFO Full at Frame Start	R

4.4.110 etherStatsOctets_P[n] — PORT[n] MAC All Frame Octets Register (n = 0..4)

Address: 4405 08B0h + 400h × n

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	etherStatsOctets_P[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	etherStatsOctets_P[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.112 etherStatsOctets_P[n] Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	etherStatsOctets_P[n]	PORT[n], this field indicates the octets of MAC All Frames, Good and Bad (incl. too long/short)	R

4.4.111 etherStatsPkts_P[n] — PORT[n] MAC All Frame Count Register (n = 0..4)

Address: 4405 08B4h + 400h × n

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	etherStatsPkts_P[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	etherStatsPkts_P[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.113 etherStatsPkts_P[n] Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	etherStatsPkts_P[n]	PORT[n], this field indicates the number of MAC All Frames, Good and Bad (incl. too long/short)	R

4.4.112 etherStatsUndersizePkts_P[n] — PORT[n] MAC Too Short Frame Count Register (n = 0..4)

Address: 4405 08B8h + 400h × n

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	etherStatsUndersizePkts_P[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	etherStatsUndersizePkts_P[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.114 etherStatsUndersizePkts_P[n] Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	etherStatsUndersizePkts_P[n]	PORT[n], this field indicates the number of MAC too Short, Good CRC	R

4.4.113 etherStatsOversizePkts_P[n] — PORT[n] MAC Too Long Frame Count Register (n = 0..4)

Address: 4405 08BCh + 400h × n

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	etherStatsOversizePkts_P[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	etherStatsOversizePkts_P[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.115 etherStatsOversizePkts_P[n] Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	etherStatsOversizePkts_P[n]	PORT[n], this field indicates the number of MAC too Long, Good CRC	R

4.4.114 etherStatsPkts64Octets_P[n] — PORT[n] MAC 64 Octets Frame Count Register (n = 0..4)

Address: 4405 08C0h + 400h × n

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	etherStatsPkts64Octets_P[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	etherStatsPkts64Octets_P[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.116 etherStatsPkts64Octets_P[n] Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	etherStatsPkts64Octets_P[n]	PORT[n], this field indicates the number of MAC All Frames, Good and Bad (Packet Size: 64 bytes)	R

4.4.115 etherStatsPkts65to127Octets_P[n] — PORT[n] MAC 65 to 127 Octets Frame Count Register (n = 0..4)

Address: 4405 08C4h + 400h × n

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	etherStatsPkts65to127Octets_P[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	etherStatsPkts65to127Octets_P[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.117 etherStatsPkts65to127Octets_P[n] Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	etherStatsPkts65to127Octets_P[n]	PORT[n], this field indicates the number of MAC All Frames, Good and Bad (Packet Size: 65 to 127 bytes)	R

4.4.116 etherStatsPkts128to255Octets_P[n] — PORT[n] MAC 128 to 255 Octets Frame Count Register (n = 0..4)

Address: 4405 08C8h + 400h × n

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	etherStatsPkts128to255Octets_P[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	etherStatsPkts128to255Octets_P[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.118 etherStatsPkts128to255Octets_P[n] Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	etherStatsPkts128to255Octets_P[n]	PORT[n], this field indicates the number of MAC All Frames, Good and Bad (Packet Size: 128 to 255 bytes)	R

4.4.117 etherStatsPkts256to511Octets_P[n] — PORT[n] MAC 256 to 511 Octets Frame Count Register (n = 0..4)

Address: 4405 08CCh + 400h × n

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	etherStatsPkts256to511Octets_P[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	etherStatsPkts256to511Octets_P[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.119 etherStatsPkts256to511Octets_P[n] Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	etherStatsPkts256to511Octets_P[n]	PORT[n], this field indicates the number of MAC All Frames, Good and Bad (Packet Size: 256 to 511 bytes)	R

4.4.118 etherStatsPkts512to1023Octets_P[n] — PORT[n] MAC 512 to 1023 Octets Frame Count Register (n = 0..4)

Address: 4405 08D0h + 400h × n

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	etherStatsPkts512to1023Octets_P[n]															
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	etherStatsPkts512to1023Octets_P[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.120 etherStatsPkts512to1023Octets_P[n] Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	etherStatsPkts512to1023Octets_P[n]	PORT[n], this field indicates the number of MAC All Frames, Good and Bad (Packet Size: 512 to 1023 bytes)	R

4.4.119 etherStatsPkts1024to1518Octets_P[n] — PORT[n] MAC 1024 to 1519 Octets Frame Count Register (n = 0..4)

Address: 4405 08D4h + 400h × n

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	etherStatsPkts1024to1518Octets_P[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	etherStatsPkts1024to1518Octets_P[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.121 etherStatsPkts1024to1518Octets_P[n] Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	etherStatsPkts1024to 1518Octets_P[n]	PORT[n], this field indicates the number of MAC All Frames, Good and Bad (Packet Size: 1024 to 1518 bytes)	R

4.4.120 etherStatsPkts1519toXOctets_P[n] — PORT[n] MAC Over 1519 Octets Frame Count Register (n = 0..4)

Address: 4405 08D8h + 400h × n

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	etherStatsPkts1519toXOctets_P[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	etherStatsPkts1519toXOctets_P[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.122 etherStatsPkts1519toXOctets_P[n] Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	etherStatsPkts1519to XOctets_P[n]	PORT[n], this field indicates the number of MAC all Frames, Good and Bad (Packet Size: 1519 bytes to TX_IPG_LENGTH_P[n])	R

4.4.121 etherStatsJabbers_P[n] — PORT[n] MAC Jabbers Frame Count Register (n = 0..4)

Address: 4405 08DCh + 400h × n

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	etherStatsJabbers_P[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	etherStatsJabbers_P[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.123 etherStatsJabbers_P[n] Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	etherStatsJabbers_P[n]	PORT[n], this field indicates the number of MAC too Long, Bad CRC	R

4.4.122 etherStatsFragments_P[n] — PORT[n] MAC Fragment Frame Count Register (n = 0..4)

Address: 4405 08E0h + 400h × n

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	etherStatsFragments_P[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	etherStatsFragments_P[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.124 etherStatsFragments_P[n] Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	etherStatsFragments_P[n]	PORT[n], this field indicates the number of MAC too Short, Bad CRC	R

4.4.123 VLANReceivedOK_P[n] — PORT[n] MAC Received VLAN Tagged Frame Count Register (n = 0..4)

Address: 4405 08E8h + 400h × n

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	VLANReceivedOK_P[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	VLANReceivedOK_P[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.125 VLANReceivedOK_P[n] Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	VLANReceivedOK_P[n]	PORT[n], this field indicates the number of Good Frames with VLAN Tag Received	R

4.4.124 VLANTransmittedOK_P[n] — PORT[n] MAC Transmitted VLAN Tagged Frame Count Register (n = 0..4)

Address: 4405 08F4h + 400h × n

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	VLANTransmittedOK_P[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	VLANTransmittedOK_P[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.126 VLANTransmittedOK_P[n] Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	VLANTransmittedOK_P[n]	PORT[n], this field indicates the number of Good Frames with VLAN Tag Transmitted	R

4.4.125 FramesRetransmitted_P[n] — PORT[n] MAC Retransmitted Frame Count Register (n = 0..4)

NOTE

- If the Hub mode retransmit is disabled (see **Section 4.4.183, HUB_CONFIG — HUB Configuration Register**), the counter increments when a collision occurred that normally would cause a retransmit, although the frame was discarded.
- The counter increments with the next frame transmitted, in case the frame was discarded due to too many retransmits or hub retransmit disabled.

Address: 4405 08F8h + 400h × n

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	FramesRetransmitted_P[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	FramesRetransmitted_P[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.127 FramesRetransmitted_P[n] Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	FramesRetransmitted_P[n]	PORT[n], this field indicates the number of Transmitted Frames that Experienced a Collision and were Retransmitted Increments once per frame, independent of number of retransmit attempts (will increment even if frame was discarded after 16 attempts). Does not increment for late collisions.	R

4.4.126 STATS_HIWORD_P[n] — PORT[n] MAC Statistics Counter High Word Register (n = 0..4)

This register exists once only and is global. Accessing it in any of the MAC pages reads from this same global register. It also means that it is latching for every statistics counter read in any of the MAC pages.

Address: 4405 0900h + 400h × n

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	STATS_HIWORD_P[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	STATS_HIWORD_P[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.128 STATS_HIWORD_P[n] Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	STATS_HIWORD_P[n]	The latched upper 32-bit of the 64 bits MAC Statistics Counter Last Read Returns latched bits 63:32 of the last accessed counter of any MAC page.	R

4.4.127 STATS_CTRL_P[n] — PORT[n] MAC Statistics Control Register (n = 0..4)

Address: 4405 0904h + 400h × n

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ClearBusy	ClearALL
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	0	0

Table 4.129 STATS_CTRL_P[n] Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b2	Reserved	Set to zero on Write. ignore on Read.	R
b1	ClearBusy	Clear in Progress Indication (RO) The bit is set when a clear command has been triggered (either by ClearAll control bit write, or soft reset write to COMMAND_CONFIG_P[n] register. The bit clears when all counters have been initialized.	R
b0	ClearALL	Self Clearing Counter Initialize Command The bit resets itself immediately, hence always reads back as 0. When written with 1, all statistics counters of all MACs will be initialized to the value given in STATS_CLEAR_VALUELO/HI. For test purposes, the clear value is programmable to perform counter wrap around or other memory tests for the counters. They should be set to all zero during normal operation. ***Note)*** A clear takes many switch system clock cycles (up to 30 * ports).	R/W

4.4.128 STATS_CLEAR_VALUELO_P[n] — PORT[n] MAC Statistics Clear Value Lower Register (n = 0..4)

Address: 4405 0908h + 400h × n

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	STATS_CLEAR_VALUELO_P[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	STATS_CLEAR_VALUELO_P[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.130 STATS_CLEAR_VALUELO_P[n] Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	STATS_CLEAR_VALUELO_P[n]	PORT[n] Lower 32-bit of 64 bits Value Loaded into All Counters when Clearing All Counters with STATS_CTRL_P[n].ClearAll command for test purposes. Should be 0 normally.	R/W

4.4.129 STATS_CLEAR_VALUEHI_P[n] — PORT[n] MAC Statistics Clear Value Higher Register (n = 0..4)

Address: 4405 090Ch + 400h × n

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	STATS_CLEAR_VALUEHI_P[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	STATS_CLEAR_VALUEHI_P[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.131 STATS_CLEAR_VALUEHI_P[n] Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	STATS_CLEAR_VALUEHI_P[n]	PORT[n] Upper 32-bit of 64 bits Value Loaded into All Counters when Clearing All Counters with STATS_CTRL_P[n].ClearAll command for test purposes. Should be 0 normally.	R/W

4.4.130 aDeferred_P[n] — PORT[n] MAC Deferred Count Register (n = 0..3)

Address: 4405 0910h + 400h × n

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	aDeferred_P[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	aDeferred_P[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.132 aDeferred_P[n] Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	aDeferred_P[n]	PORT[n], this field indicates the number of Frame Transmitted without Collision but was Deferred at Begin	R

4.4.131 aMultipleCollisions_P[n] — PORT[n] MAC Multiple Collision Count Register (n = 0..3)

Address: 4405 0914h + 400h × n

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	aMultipleCollisions_P[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	aMultipleCollisions_P[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.133 aMultipleCollisions_P[n] Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	aMultipleCollisions_P[n]	PORT[n], this field indicates the number of Good Frame Transmit after Multiple Collisions	R

4.4.132 aSingleCollisions_P[n] — PORT[n] MAC Single Collision Count Register (n = 0..3)

Address: 4405 0918h + 400h × n

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	aSingleCollisions_P[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	aSingleCollisions_P[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.134 aSingleCollisions_P[n] Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	aSingleCollisions_P[n]	PORT[n], this field indicates the number of Good Frame Transmit after Single Collision	R

4.4.133 aLateCollisions_P[n] — PORT[n] MAC Late Collision Count Register (n = 0..3)

Address: 4405 091Ch + 400h × n

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	aLateCollisions_P[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	aLateCollisions_P[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.135 aLateCollisions_P[n] Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	aLateCollisions_P[n]	PORT[n], this field indicates the number of too Late Collision. Frame was Aborted and not Retransmitted	R

4.4.134 aExcessiveCollisions_P[n] — PORT[n] MAC Excessive Collision Count Register (n = 0..3)

Address: 4405 0920h + 400h × n

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	aExcessiveCollisions_P[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	aExcessiveCollisions_P[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.136 aExcessiveCollisions_P[n] Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	aExcessiveCollisions_P[n]	PORT[n], this field indicates the number of Frames Discarded due to 16 Consecutive Collisions	R

4.4.135 aCarrierSenseErrors_P[n] — PORT[n] MAC Carrier Sense Error Count Register (n = 0..3)

Address: 4405 0924h + 400h × n

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	aCarrierSenseErrors_P[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	aCarrierSenseErrors_P[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.137 aCarrierSenseErrors_P[n] Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	aCarrierSenseErrors_P[n]	PORT[n] Increments when during Transmission without Collisions the PHY Carrier Sense Signal (RX_CRS) Dropped or never Asserted	R

4.4.136 DLR_CONTROL — DLR Control Register

Address: 4405 3C00h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	US_TIME			
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	US_TIME								—	—	—	IGNORE_INVTM	—	—	AUTOFLUSH	ENABLE
Value after reset	0	1	1	0	0	1	0	0	X	X	X	0	X	X	0	0

Table 4.138 DLR_CONTROL Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b20	Reserved	Set to zero on Write. ignore on Read.	R
b19 to b8	US_TIME	Number of clock cycles required for 1 microsecond for the switch system clock. Default value assumes in case of running on 100 MHz. Should be changed according to the current switch system clock frequency. ***Example*:** If the switch system clock frequency is 125 MHz, should be set 125.	R/W
b7 to b5	Reserved	Set to zero on Write. ignore on Read.	R
b4	IGNORE_INVTM	Enable ignore Beacon frames with invalid timeout timer. When enabled (1), frames with timeout timer value not within a range of 200 microseconds to 500 milliseconds will be ignored and parameters will not be locally stored or considered for state transitions. The timeout timer value will always be stored within the register DLR_INV_TMOUT irrespective of the value of this bit. Ignored frames will be forwarded normally.	R/W
b3, b2	Reserved	Set to zero on Write. ignore on Read.	R
b1	AUTOFLUSH	Enable automatic flushing of unicast entries in address table if ring reconfiguration occurs (see also IRQ_flush_macaddr_ena bit of DLR_IRQ_CONTROL register). When set and a table flush event occurs, it executes the lookup engine command LK_ADDR_CTRL.DELETE_PORT with a mask of 0x03. This deletes all dynamic entries in the table that are listed for forwarding to ports 0, 1 (i.e. it deletes all learned unicast that are located within the ring).	R/W
b0	ENABLE	Enable DLR extension module. When set, the DLR module becomes active. This also enables the loop filter. When cleared, no DLR frame detection occurs and all DLR module statistics registers are cleared.	R/W

4.4.137 DLR_STATUS — DLR Status Register

Address: 4405 3C04h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TOPOLOGY								—	—	—	—	—	—	LINK_STATUS	
Value after reset	0	0	0	0	0	0	0	0	X	X	X	X	X	X	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	NODE_STATE								—	—	—	—	—	—	LastBcnRcvPort	
Value after reset	0	0	0	0	0	0	0	0	X	X	X	X	X	X	0	0

Table 4.139 DLR_STATUS Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b24	TOPOLOGY	Current Network Topology 0x00: Linear Topology when local node is in IDLE state 0x01: Ring Topology when local node is not in IDLE state	R
b23 to b18	Reserved	Set to zero on Write. ignore on Read.	R
b17, b16	LINK_STATUS	Link Status Bit 16: Set if port 0 link is up. Bit 17: Set if port 1 link is up. ***Note*)** This is a direct representation of the Link status input from PHY.	R
b15 to b8	NODE_STATE	Local Node Current State 0x00: IDLE_STATE 0x01: NORMAL_STATE 0x02: FAULT_STATE Others: unused	R
b7 to b2	Reserved	Set to zero on Write. ignore on Read.	R
b1, b0	LastBcnRcvPort	Last Beacon Receive Port Bit 0: Set if beacon frame from active supervisor received on port 0 Bit 1: Set if beacon frame from active supervisor received on port 1	R

4.4.138 DLR_ETH_TYP — DLR Ethernet Type Register

Address: 4405 3C08h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DLR_ETH_TYP															
Value after reset	1	0	0	0	0	0	0	0	1	1	1	0	0	0	0	1

Table 4.140 DLR_ETH_TYP Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b16	Reserved	Set to zero on Write. ignore on Read.	R
b15 to b0	DLR_ETH_TYP	Ethernet Type for DLR Frame Detection	R/W

4.4.139 DLR_IRQ_CONTROL — DLR Interrupt Control Register

Address: 4405 3C0Ch

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	atomic_AND	atomic_OR	low_int_en	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	IRQ_frm_dscrd1	IRQ_frm_dscrd0	IRQ_bec_rcv1_ena	IRQ_bec_rcv0_ena	IRQ_invalid_tmr_ena	IRQ_ip_addr_chng_ena	IRQ_sup_ignord_ena	IRQ_link_chng1_ena	IRQ_link_chng0_ena	IRQ_supr_chng_ena	IRQ_bec_tmr1_exp_ena	IRQ_bec_tmr0_exp_ena	IRQ_stop_nbchk1_ena	IRQ_stop_nbchk0_ena	IRQ_flush_macaddr_ena	IRQ_state_chng_ena
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.141 DLR_IRQ_CONTROL Register Contents

Bit Position	Bit Name	Function	R/W
b31	atomic_AND	When set during a register write, the enable bits are AND'ed with the current setting of the register (i.e. clearing bits).	R/W
b30	atomic_OR	When set during a register write, the enable bits are OR'ed into the current setting of the register (i.e. setting bits).	R/W
b29	low_int_en	Enable active low interrupt. Asserted to use active low interrupt signal instead of active high interrupt signal.	R/W
b28 to b16	Reserved	Set to zero on Write. ignore on Read.	R
b15	IRQ_frm_dscrd1	Enable Interrupt on Frame discard due to source address match with the local address on Port 1.	R/W
b14	IRQ_frm_dscrd0	Enable Interrupt on Frame discard due to source address match with the local address on Port 0.	R/W
b13	IRQ_bec_rcv1_ena	Enable Interrupt on Beacon frame detection on Port 1.	R/W
b12	IRQ_bec_rcv0_ena	Enable Interrupt on Beacon frame detection on Port 0.	R/W
b11	IRQ_invalid_tmr_ena	Enable Interrupt on invalid range for beacon timeout timer value detection.	R/W
b10	IRQ_ip_addr_chng_ena	Enable interrupt on IP address change detection within beacon frame from ring supervisor.	R/W
b9	IRQ_sup_ignord_ena	Enable interrupt on beacon frame detection from a supervisor with lower precedence than the current ring supervisor or lower numeric value for MAC address when precedence is same.	R/W
b8	IRQ_link_chng1_ena	Enable Link change interrupt event for Port 1.	R/W
b7	IRQ_link_chng0_ena	Enable Link change interrupt event for Port 0.	R/W
b6	IRQ_supr_chng_ena	Enable interrupt on ring supervisor change.	R/W
b5	IRQ_bec_tmr1_exp_ena	Enable interrupt on beacon timeout timer expire for port 1.	R/W
b4	IRQ_bec_tmr0_exp_ena	Enable interrupt on beacon timeout timer expire for port 0.	R/W
b3	IRQ_stop_nbchk1_ena	Enable Stop for neighbor check timeout timer interrupt for port 1.	R/W
b2	IRQ_stop_nbchk0_ena	Enable Stop for neighbor check timeout timer interrupt for port 0.	R/W
b1	IRQ_flush_macaddr_ena	Enable flush local MAC address table interrupt. When enabled, an interrupt is generated when unicast MAC address learning table should be flushed.	R/W
b0	IRQ_state_chng_ena	Enable interrupt for state change. When enabled, an interrupt is generated when state change occurs for the local beacon based DLR ring node.	R/W

4.4.140 DLR_IRQ_STAT_ACK — DLR Interrupt Status/ACK Register

Address: 4405 3C10h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	frm_dscrd1_IRQ_pending	frm_dscrd0_IRQ_pending	bec_rcv1_IRQ_pending	bec_rcv0_IRQ_pending	invalid_tmr_IRQ_pending	ip_chng_IRQ_pending	sup_ignord_IRQ_pending	Link1_IRQ_pending	Link0_IRQ_pending	supr_chng_IRQ_pending	bec_tmr1_IRQ_pending	bec_tmr0_IRQ_pending	nbchk1_IRQ_pending	nbchk0_IRQ_pending	flush_IRQ_pending	state_chng_IRQ_pending
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.142 DLR_IRQ_STAT_ACK Register Contents (1/2)

Bit Position	Bit Name	Function	R/W
b31 to b16	Reserved	Set to zero on Write. ignore on Read.	R
b15	frm_dscrd1_IRQ_pending	Latched event on Frame discard due to source address match with the local address on Port 1 (loop filter). To clear the latch, the bit must be written with “1”. ***Note*)** The latch operates independently from the enable bit 15 of DLR_IRQ_CONTROL.	R/W
b14	frm_dscrd0_IRQ_pending	Latched event on Frame discard due to source address match with the local address on Port 0 (loop filter). To clear the latch, the bit must be written with “1”. ***Note*)** The latch operates independently from the enable bit 14 of DLR_IRQ_CONTROL.	R/W
b13	bec_rcv1_IRQ_pending	Latched event on Beacon frame detection on Port 1. To clear the latch, the bit must be written with “1”. ***Note*)** The latch operates independently from the enable bit 13 of DLR_IRQ_CONTROL.	R/W
b12	bec_rcv0_IRQ_pending	Latched event on Beacon frame detection on Port 0. To clear the latch, the bit must be written with “1”. ***Note*)** The latch operates independently from the enable bit 12 of DLR_IRQ_CONTROL.	R/W
b11	invalid_tmr_IRQ_pending	Latched event on invalid beacon timeout timer value detection within beacon frame on port 0 or port 1. To clear the latch, the bit must be written with “1”. ***Note*)** The latch operates independently from the enable bit 11 of DLR_IRQ_CONTROL.	R/W
b10	ip_chng_IRQ_pending	Latched IP address change event. To clear the latch, the bit must be written with “1”. ***Note*)** The latch operates independently from the enable bit 10 of DLR_IRQ_CONTROL.	R/W
b9	sup_ignord_IRQ_pending	Latched event for beacon frame detection from ignored supervisor. Asserted when Beacon frame detected from a supervisor with lower precedence or lower numeric value for the MAC address when precedence is same. To clear the latch, the bit must be written with “1”. ***Note*)** The latch operates independently from the enable bit 9 of DLR_IRQ_CONTROL.	R/W
b8	Link1_IRQ_pending	Latched link status change event. Asserts whenever link status change is detected on Port 1. To clear the latch, the bit must be written with “1”. ***Note*)** The latch operates independently from the enable bit 8 of DLR_IRQ_CONTROL.	R/W

Table 4.142 DLR_IRQ_STAT_ACK Register Contents (2/2)

Bit Position	Bit Name	Function	R/W
b7	Link0_IRQ_pending	Latched link status change event. Asserts whenever link status change is detected on Port 0. To clear the latch, the bit must be written with “1”. ***Note*)** The latch operates independently from the enable bit 7 of DLR_IRQ_CONTROL.	R/W
b6	supr_chng_IRQ_pending	Latched supervisor change event. Asserted, when ring supervisor changes for DLR ring. To clear the latch, the bit must be written with “1”. ***Note*)** The latch operates independently from the enable bit 6 of DLR_IRQ_CONTROL.	R/W
b5	bec_tmr1_IRQ_pending	Beacon timeout timer expire interrupt for port 1. Asserted when Beacon timeout timer expired for port 1. To clear the latch, the bit must be written with “1”. ***Note*)** The latch operates independently from the enable bit 5 of DLR_IRQ_CONTROL.	R/W
b4	bec_tmr0_IRQ_pending	Beacon timeout timer expire interrupt for port 0. Asserted when Beacon timeout timer expired for port 0. To clear the latch, the bit must be written with “1”. ***Note*)** The latch operates independently from the enable bit 4 of DLR_IRQ_CONTROL.	R/W
b3	nbchk1_IRQ_pending	Stop event for neighbor check timeout timer for port 1. When asserted, neighbor check timeout timer should be stopped. To clear the latch, the bit must be written with “1”. ***Note*)** The latch operates independently from the enable bit 3 of DLR_IRQ_CONTROL.	R/W
b2	nbchk0_IRQ_pending	Stop event for neighbor check timeout timer for port 0. When asserted, neighbor check timeout timer should be stopped. To clear the latch, the bit must be written with “1”. ***Note*)** The latch operates independently from the enable bit 2 of DLR_IRQ_CONTROL.	R/W
b1	flush_IRQ_pending	Latched flush event for MAC address learning table. When asserted, MAC address learning table should be flushed. To clear the latch, the bit must be written with “1”. ***Note*)** The latch operates independently from the enable bit 1 of DLR_IRQ_CONTROL.	R/W
b0	state_chng_IRQ_pending	Latched state change event. Asserts whenever local beacon based DLR node state change. To clear the latch, the bit must be written with “1”. ***Note*)** The latch operates independently from the enable bit 0 of DLR_IRQ_CONTROL.	R/W

4.4.141 DLR_LOC_MAClo — DLR Local MAC Address Low Register

Address: 4405 3C14h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	LOC_MAC															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	LOC_MAC															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.143 DLR_LOC_MAClo Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	LOC_MAC	First 4 Octets of the Local MAC Address for Loop Filter Bits 7:0 is first byte, ..., Bits 31:24 is 4th byte of address	R/W

4.4.142 DLR_LOC_MAChi — DLR Local MAC Address High Register

Address: 4405 3C18h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	LOC_MAC															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.144 DLR_LOC_MAChi Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b16	Reserved	Set to zero on Write. ignore on Read.	R
b15 to b0	LOC_MAC	Last 2 Octets of Local MAC Address for Loop Filter Bits 7:0 is 5th byte of the local MAC address. Bits 15:8 is 6th (last) byte of the local MAC address	R/W

4.4.143 DLR_SUPR_MAClo — DLR Supervisor MAC Address Low Register

Address: 4405 3C20h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	SUPR_MAC															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SUPR_MAC															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.145 DLR_SUPR_MAClo Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	SUPR_MAC	First 4 Octets of the Active Ring Supervisor's MAC Address Extracted from Source Address Field of the Beacon Frame Bits 7:0 is first byte, ..., Bits 31:24 is 4th byte of address	R

4.4.144 DLR_SUPR_MAChi — DLR Supervisor MAC Address High Register

Address: 4405 3C24h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	PRECE							
Value after reset	X	X	X	X	X	X	X	X	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SUPR_MAC															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.146 DLR_SUPR_MAChi Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b24	Reserved	Set to zero on Write. ignore on Read.	R
b23 to b16	PRECE	Bits 23:16 is ring supervisor's precedence value extracted from the Supervisor precedence field of the Beacon frame.	R
b15 to b0	SUPR_MAC	Last 2 octets of active ring supervisor's MAC address extracted from the Source Address field of the Beacon frame. Bits 7:0 is 5th byte, 15:8 is 6th byte.	R

4.4.145 DLR_STATE_VLAN — DLR Ring Status/VLAN Register

Address: 4405 3C28h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	VLANTAG															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	VLANVALID	RINGSTAT							
Value after reset	X	X	X	X	X	X	X	0	0	0	0	0	0	0	0	0

Table 4.147 DLR_STATE_VLAN Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b16	VLANTAG	Bits 31:16 is 802.1Q VLAN Tag control field extracted from the VLAN info field of the Beacon frame.	R
b15 to b9	Reserved	Set to zero on Write. ignore on Read.	R
b8	VLANVALID	VLAN Valid When asserted, current VLAN Tag control field (Bits 31:16) contains valid VLAN ID.	R
b7 to b0	RINGSTAT	DLR ring state extracted from the Ring State field of the Beacon frame. 0x1: RING_NORMAL_STATE 0x2: RING_FAULT_STATE Others: Unused	R

4.4.146 DLR_BEC_TMOUT — DLR Beacon Timeout Register

Address: 4405 3C2Ch

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	BEC_TMOUT															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	BEC_TMOUT															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.148 DLR_BEC_TMOUT Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	BEC_TMOUT	Beacon timeout timer value extracted from the Beacon Timeout in microseconds field of the Beacon frame. Invalid timeout timer values will be not written to this register and will not be considered for state transition when ignore enabled through the bit 8 of the CONTROL register. Expected value is in a range between minimum 200 microseconds and maximum 500 milliseconds. Typical value is 1960 microseconds.	R

4.4.147 DLR_BEC_INTRVL — DLR Beacon Interval Register

Address: 4405 3C30h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	BEC_INTRVL															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	BEC_INTRVL															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.149 DLR_BEC_INTRVL Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	BEC_INTRVL	Beacon interval extracted from the Beacon Interval field of the Beacon frame. Expected value is in a range between minimum 100 microseconds and maximum 100 milliseconds. Typical value is 400 microseconds.	R

4.4.148 DLR_SUPR_IPADR — DLR Supervisor IP Address Register

Address: 4405 3C34h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	SUPR_IPADR															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SUPR_IPADR															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.150 DLR_SUPR_IPADR Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	SUPR_IPADR	Ring supervisor's IP address extracted from the Source IP address field of the Beacon frame. A value of 0x0 can be received when supervisor has no IP address.	R

4.4.149 DLR_ETH_STYP_VER — DLR Sub Type/Protocol Version Register

Address: 4405 3C38h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	SPORT							
Value after reset	X	X	X	X	X	X	X	X	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PROTVER								SUBTYPE							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.151 DLR_ETH_STYP_VER Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b24	Reserved	Set to zero on Write. ignore on Read.	R
b23 to b16	SPORT	Source Port Extracted from the Source Port Field of the Beacon frame	R
b15 to b8	PROTVER	DLR Ring Protocol Version Extracted from the Ring Protocol Version Field of the Beacon Frame	R
b7 to b0	SUBTYPE	DLR Ring Ether Sub Type Extracted from the Ring Sub Type Field of the Beacon Frame Value is expected to be 0x02.	R

4.4.150 DLR_INV_TMOUT — DLR Beacon Timeout Timer Register

Address: 4405 3C3Ch

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	INV_TMOUT															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	INV_TMOUT															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.152 DLR_INV_TMOUT Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	INV_TMOUT	Last out of Range Beacon Timeout Timer Value Extracted from Beacon Frame on any of the Port Valid range is between 200 microseconds and 500 milliseconds. This register will be always written whenever new invalid value is received. Contains a valid value when the bit 11 of the register DLR_IRQ_STAT_ACK is asserted.	R

4.4.151 DLR_SEQ_ID — DLR Sequence ID Register

Address: 4405 3C40h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	SEQ_ID															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SEQ_ID															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.153 DLR_SEQ_ID Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	SEQ_ID	Sequence ID of the last Beacon frame extracted from the Sequence ID field of the Beacon frame on port 0 or port 1. Sequence ID of the ignored frames is not stored.	R

4.4.152 DLR_DSTlo — DLR Beacon Destination Address Low Register

Address: 4405 3C58h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	DLR_DST															
Value after reset	0	0	0	0	0	0	0	0	0	1	1	0	1	1	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DLR_DST															
Value after reset	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	1

Table 4.154 DLR_DSTlo Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	DLR_DST	First 4 Octets of the Beacon Frame Destination Multicast Address (01-21-6C-00-00-01) Bits 7:0 is first byte, ..., Bits 31:24 is 4th byte of address	R/W

4.4.153 DLR_DSThi — DLR Beacon Destination Address High Register

Address: 4405 3C5Ch

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DLR_DST															
Value after reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

Table 4.155 DLR_DSThi Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b16	Reserved	Set to zero on Write. ignore on Read.	R
b15 to b0	DLR_DST	Last 2 Octets of the Beacon Frame Destination Multicast Address (01-21-6C-00-00-01) Bits 7:0 is 5th byte, 15:8 is 6th.	R/W

4.4.154 DLR_RX_STAT0 — DLR Received Frame Statistic Register 0

Address: 4405 3C60h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	RX_STAT0															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RX_STAT0															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.156 DLR_RX_STAT0 Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	RX_STAT0	Number of Beacon Frames Received on Port 0 Counter will increment with the Beacon frames which matches the destination MAC address, Ether type, DLR frame type and CRC. Ignored frames due to type mismatch or CRC error are not counted. The counters are cleared if the DLR module is disabled.	R

4.4.155 DLR_RX_ERR_STAT0 — DLR Received Frame Error Statistic Register 0

Address: 4405 3C64h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	RX_ERR_STAT0															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RX_ERR_STAT0															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.157 DLR_RX_ERR_STAT0 Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	RX_ERR_STAT0	Number of Beacon Frames Received with CRC Error on Port 0 Counter will increment with the Beacon frames which match the destination MAC address, Ether type, DLR frame type but with CRC error. The counters are cleared if the DLR module is disabled.	R

4.4.156 DLR_TX_STAT0 — DLR Transmitted Frame Statistic Register 0

Address: 4405 3C68h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TX_STAT0															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TX_STAT0															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.158 DLR_TX_STAT0 Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	TX_STAT0	Number of Beacon Frames Forwarded from Port 1 to Port 0 (Port 0 Transmitted) The counters are cleared if the DLR module is disabled. ***Note)*** Not available, always 0.	R

4.4.157 DLR_RX_STAT1 — DLR Received Frame Statistic Register 1

Address: 4405 3C70h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	RX_STAT1															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RX_STAT1															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.159 DLR_RX_STAT1 Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	RX_STAT1	Number of Beacon Frames Received on Port 1 Counter will increment with the Beacon frames which matches the destination MAC address, Ether type, DLR frame type and CRC. Ignored frames due to type mismatch or CRC error are not counted. The counters are cleared if the DLR module is disabled.	R

4.4.158 DLR_RX_ERR_STAT1 — DLR Received Frame Error Statistic Register 1

Address: 4405 3C74h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	RX_ERR_STAT1															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RX_ERR_STAT1															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.160 DLR_RX_ERR_STAT1 Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	RX_ERR_STAT1	Number of Beacon Frames Received with CRC Error on Port 1 Counter will increment with the Beacon frames which matches the destination MAC address, Ether type, DLR frame type but with CRC error. The counters are cleared if the DLR module is disabled.	R

4.4.159 DLR_TX_STAT1 — DLR Transmitted Frame Statistic Register 1

Address: 4405 3C78h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TX_STAT1															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TX_STAT1															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.161 DLR_TX_STAT1 Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	TX_STAT1	Number of Beacon Frames Forwarded from Port 0 to Port 1 (Port 1 Transmitted) The counters are cleared if the DLR module is disabled. ***Note*)** Not available, always 0.	R

4.4.160 PRP_CONFIG — PRP Configuration Register

Address: 4405 3D00h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PRP_AGE_ENA
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	TX_RCT_1588	TX_RCT_UNKNOWN	TX_RCT_MULTICAST	TX_RCT_BROADCAST	TX_RCT_MODE		RX_REMOVE_RCT	RX_DUP_ACCEPT	PRP_ENA
Value after reset	X	X	X	X	X	X	X	0	0	0	0	0	0	0	0	0

Table 4.162 PRP_CONFIG Register Contents (1/2)

Bit Position	Bit Name	Function	R/W
b31 to b17	Reserved	Set to zero on Write. ignore on Read.	R
b16	PRP_AGE_ENA	Enable History Memory Aging Timer The timer value is configured with register PRP_AGETIME.	R/W
b15 to b9	Reserved	Set to zero on Write. ignore on Read.	R
b8	TX_RCT_1588	1: Allow appending RCT to IEEE 1588 frames (ether type 0x88f7). Defines to treat those frames as any other frame according to the configured TX_RCT_xxx settings below. 0: Do not append RCT to IEEE 1588 frames. Even if they are duplicated. This affects IEEE 1588 frames that are forwarded through the switch (e.g. when used as RedBox) to both PRP_GROUP ports. Locally generated IEEE 1588 frames (e.g. peer delay request/response) are not affected by this setting. ***Note*)** Should be 0 normally, as IEEE 1588 frames should not be considered for duplicate detection (IEC 62439-3:A.4.4.).	R/W
b7	TX_RCT_UNKNOWN	1: Allow appending RCT to frames that have unknown destination and are hence flooded. 0: Do not append RCT to frames flooded because the destination is unknown. This further restricts modes 00b, 01b, 11b. ***Note*)** Should be 1 normally.	R/W
b6	TX_RCT_MULTICAST	1: Allow appending RCT to multicast frames. 0: Do not append RCT to multicast frames. This further restricts modes 00b, 01b, 11b. ***Note*)** Should be 1 normally.	R/W
b5	TX_RCT_BROADCAST	1: Allow appending RCT to broadcast frames. 0: Do not append RCT to broadcast frames. This further restricts modes 00b, 01b, 11b. ***Note*)** Should be 1 normally.	R/W

Table 4.162 PRP_CONFIG Register Contents (2/2)

Bit Position	Bit Name	Function	R/W
b4, b3	TX_RCT_MODE	Control Appending the RCT to Transmitted Frames on the Redundant Ports The following modes are available: [4:3] 00b: Append RCT if group only: Append RCT, if frame is duplicated to exactly the ports listed in the PRP_GROUP register. If the frame is sent to only one of the ports, or is e.g. flooded also to other ports no RCT is appended. 01b: Append RCT to group always: Append the RCT always if the frame is duplicated to at least the two ports listed in the PRP_GROUP. This includes flooded frames even if they are additionally sent on other ports outside the PRP_GROUP. If the frame is sent to only one of the PRP_GROUP ports, no RCT is appended. 10b: Disable RCT append. The transmitters never append the RCT even if frames are duplicated to the PRP_GROUP ports. (This represents the IEC 62439-3:4.2.6 duplicate accept mode of operation) 11b: Append RCT always forced. The transmitters will append the RCT to all frames transmitted on any of the PRP_GROUP ports, no matter if they are duplicated or not or if they are flooded. Normally the setting should be 00b or 01b. This allows Single attached nodes (SAN) and Double attached nodes (DAN) within the PRP LAN segments ensuring the RCT is only appended to frames sent to DANs. ***Note*)** BPDU frames will never get an RCT appended (IEC:4.2.7.5.1), independent from this setting. Frames forwarded to a port that is not within the PRP_GROUP will never get a RCT appended (even if at the same time the frame is duplicated to the PRP_GROUP where it can get the RCT appended).	R/W
b2	RX_REMOVE_RCT	Allow PRP Port RX to Remove the RCT When 1, the RCT is removed from frames received at the PRP ports before forwarding. Has no effect on other ports. When 0, the RCT is not removed. Duplicate detection occurs normally, independent from this setting. BPDU and IEEE 1588 Layer 2 frames are not affected by this setting and are received unmodified always. ***Note*)** The RCT is removed only if the source node is known to be a DAN as indicated by having a port mask containing both ports within the address table and the LAN ID matches the port where it is received.	R/W
b1	RX_DUP_ACCEPT	Enable Duplicate Accept Mode of Operation at Receive When 1, the receivers will not perform duplicate detection and forward all received frames unmodified. When 0, the receivers try to detect and discard duplicates. Should be 0 for normal operation. ***Note*)** Even when 0, BPDU and IEEE 1588 frames are always accepted without a duplicate check. Using receive duplicate accept could be used to perform duplicate detection at the application and not having the hardware to perform it. Aside from this use, duplicate accept mode is a test setting (IEC:4.2.6) not intended to be used during normal operation. The bit is permanent 1 (RO) in non-PRP-supported product.	R/W
b0	PRP_ENA	Enable PRP Operation All other configuration registers must have been configured correctly before setting this bit. When cleared, the statistics counters are cleared also. ***Note*)** Before re-enabling the history memory should be cleared by executing the memory RM_ADDR_CTRL.CLEAR_MEMORY command.	R/W

4.4.161 PRP_GROUP — PRP Port Group Register

Address: 4405 3D04h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	LANB_MASK			
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	PRP_GROUP			
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	0	0	0	0

Table 4.163 PRP_GROUP Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b20	Reserved	Set to zero on Write. ignore on Read.	R
b19 to b16	LANB_MASK	Defines which of the ports is considered the LAN B port. This port will have the RCT containing the LAN B identifier. The other port will use the LAN A identifier for the RCT. Exactly 1 bit must be set matching one of the two bits in the PRP_GROUP mask. Bit 16 = Port0, Bit 17 = Port1, ... ***Note)*** The setting can be changed any time during operation. For example, if management detects the ports were connected to the wrong networks it is sufficient to swap the declaration of LAN A/B in this register and there is no need to physically change connections.	R/W
b15 to b4	Reserved	Set to zero on Write. ignore on Read.	R
b3 to b0	PRP_GROUP	Defines which two ports should be treated as redundant ports for PRP. Exactly 2 bits must be set in the mask. Only line ports are supported. Bit 0 = Port 0, Bit 1 = Port 1, … ***Note)*** Setting not exactly 2 bits leads to unpredictable behavior.	R/W

4.4.162 PRP_SUFFIX — PRP RCT Suffix

Address: 4405 3D08h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PRP_SUFFIX															
Value after reset	1	0	0	0	1	0	0	0	1	1	1	1	1	0	1	1

Table 4.164 PRP_SUFFIX Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b16	Reserved	Set to zero on Write. ignore on Read.	R
b15 to b0	PRP_SUFFIX	The Redundancy Control Trailer (RCT) Suffix	R/W

4.4.163 PRP_LANID — PRP LAN Identifier

Address: 4405 3D0Ch

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	LANBID				LANAID			
Value after reset	X	X	X	X	X	X	X	X	1	0	1	1	1	0	1	0

Table 4.165 PRP_LANID Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b8	Reserved	Set to zero on Write. ignore on Read.	R
b7 to b4	LANBID	LAN B Identifier	R/W
b3 to b0	LANAID	LAN A Identifier	R/W

4.4.164 DUP_W — PRP Max Duplicate Detection Window Size

Address: 4405 3D10h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	DUP_W								
Value after reset	X	X	X	X	X	X	X	0	0	0	0	0	0	0	0	0

Table 4.166 DUP_W Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b9	Reserved	Set to zero on Write. ignore on Read.	R
b8 to b0	DUP_W	Maximum Duplicate Detect Window Size If the current frame's sequence number is in the range of expected window, it will be treated as duplicate (i.e. it is an old sequence number). If a sequence number with a value beyond this window is found, the frame is accepted and an increment of CntOutOfSeqLow occurs. Valid values are 2..255. Recommended: >50 ***Note*)** Setting values 0 or 1 will cause all frames with a sequence number less than the minimum received to be accepted with an out of sequence error indication.	R/W

4.4.165 PRP_AGETIME — PRP Aging Time Define Register

Address: 4405 3D14h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	PRP_AGETIME							
Value after reset	X	X	X	X	X	X	X	X	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PRP_AGETIME															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.167 PRP_AGETIME Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b24	Reserved	Set to zero on Write. ignore on Read.	R
b23 to b0	PRP_AGETIME	Timeout in Steps of 32 Switch system clock Cycles to Control Aging of Duplicate History Data Every Agetime one entry of the history table will be inspected for invalidating. As only a single age bit is used, it requires at maximum two table rounds until an entry is removed. The timeout should be chosen to guarantee the full table is covered in EntryForgetTime/2 (200 ms). A 24-bit value is available. A Value of 0 disables the aging process. ***Example*:** For a setting of 200 ms with a 8192 entries table and the switch system clock running at 200 MHz (5 ns): Agetime = 200 ms / (8192×5 ns) / 32 = 152.	R/W

4.4.166 PRP_IRQ_CONTROL — PRP Interrupt Control Register

If any of the interrupt enable bits is set and a corresponding event occurs, the PRP interrupt within the global INT_STAT_ACK interrupt status register asserts. In addition, the signal A5PSW_PRP_Int asserts. To clear any of the interrupts, the PRP_IRQ_STAT_ACK register must be written.

Address: 4405 3D18h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	SEQMISSING	OUTOFSEQ	WRONGLAN	MEMTOOLATE
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	0	0	0	0

Table 4.168 PRP_IRQ_CONTROL Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b4	Reserved	Set to zero on Write. ignore on Read.	R
b3	SEQMISSING	Enable interrupt for frames received and accepted that caused the history to skip a sequence number that was never received (i.e. a missing sequence number is being ignored and will now be treated as candidate for dropping). This event also caused incrementing the CntMissing statistics.	R/W
b2	OUTOFSEQ	Enable interrupt for frames received and accepted but have an unexpected sequence number. This event also caused incrementing the CntOutOfSeq statistics.	R/W
b1	WRONGLAN	This bit is not available. Set to zero on Write.	R/W
b0	MEMTOOLATE	Enable interrupt for memory error indications. When enabled, an interrupt is generated when the memory transactions for a frame could not be completed in time. For transmit, it means a frame was sent without appending an RCT. For receive, it means a RCT could not be checked and the frame was accepted. ***Note)*** This interrupt should never occur. It could occur on Gigabit ports if the switch system frequency is too low and coincidently all ports request memory lookups simultaneously. This may indicate the system frequency is too low.	R/W

4.4.167 PRP_IRQ_STAT_ACK — PRP Interrupt Status/ACK Register

To clear any of the interrupts, the PRP_IRQ_STAT_ACK register must be written.

Address: 4405 3D1Ch

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	SEQMISSING	OUTOFSEQ	WRONGLAN	MEMTOOLATE
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	0	0	0	0

Table 4.169 PRP_IRQ_STAT_ACK Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b4	Reserved	Set to zero on Write. ignore on Read.	R
b3	SEQMISSING	Interrupt pending indication bits corresponding to the bits defined in PRP_IRQ_CONTROL. If an event occurs, the corresponding bit is latched high. To clear the bit, it must be written with a “1”. ***Note)*** The latches operate independently of the PRP_IRQ_CONTROL register. That is, even when the interrupt is disabled, the event occurrences can still be inspected.	R/W
b2	OUTOFSEQ	Interrupt pending indication bits corresponding to the bits defined in PRP_IRQ_CONTROL. If an event occurs, the corresponding bit is latched high. To clear the bit, it must be written with a “1”. ***Note)*** The latches operate independently of the PRP_IRQ_CONTROL register. That is, even when the interrupt is disabled, the event occurrences can still be inspected.	R/W
b1	WRONGLAN	This bit is not available. Set to zero on Write.	R/W
b0	MEMTOOLATE	Interrupt pending indication bits corresponding to the bits defined in PRP_IRQ_CONTROL. If an event occurs, the corresponding bit is latched high. To clear the bit, it must be written with a “1”. ***Note)*** The latches operate independently of the PRP_IRQ_CONTROL register. That is, even when the interrupt is disabled, the event occurrences can still be inspected.	R/W

4.4.168 RM_ADDR_CTRL — PRP History Memory Transactions Control Register

Address: 4405 3D20h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	BUSY	—	CLEAR	—	—	READ	WRITE	—	CLEAR_MEMORY	CLEAR_DYNAMIC	—	—	—	—	—	—
Value after reset	0	X	0	X	X	0	0	X	0	0	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	address												
Value after reset	X	X	X	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.170 RM_ADDR_CTRL Register Contents

Bit Position	Bit Name	Function	R/W
b31	BUSY	Transaction Busy Indication Controller is busy. As long as the controller is busy, the corresponding command bit is set. When it becomes non busy again, all command bits are cleared. No other commands are allowed when BUSY = 1. A write to this register must be avoided as long as it is busy.	R
b30	Reserved	Set to zero on Write. ignore on Read.	R
b29	CLEAR	Writes all zero to the entry selected by the given address. Avoids the need to set the data registers to a zero value. Bit is cleared when the function has completed	R/W
b28, b27	Reserved	Set to zero on Write. ignore on Read.	R
b26	READ	When set to 1, perform single read transaction. Data is returned in RM_DATA register. Bit is cleared when the function has completed	R/W
b25	WRITE	When set to 1, perform a single write transaction. The RM_DATA register must be set prior to starting the transaction. Bit is cleared when the function has completed	R/W
b24	Reserved	Set to zero on Write. ignore on Read.	R
b23	CLEAR_MEMORY	Writes all memory locations with zero. Bit is cleared when the function has completed. ***Note)*** The bit is set upon reset to clear the memory and will deassert when completed.	R/W
b22	CLEAR_DYNAMIC	Scans the complete table for valid dynamic history entries and deletes them (writes entry with all zero). The stored sequence numbers (for transmit RCT generation) of source addresses associated with non redundant local ports are not modified. Bit is cleared when the function has completed.	R/W
b21 to b13	Reserved	Set to zero on Write. ignore on Read.	R
b12 to b0	address	Memory Address for Read and Write Transactions This is the address of an entry. The address is the same as the address of an entry in the MAC address lookup table. Hence to find the data for a specific MAC address, first the MAC address table must be searched (lookup) to retrieve the address of an entry.	R/W

4.4.169 RM_DATA — PRP Memory Data Register

Address: 4405 3D24h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	RM_DATA															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RM_DATA															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.171 RM_DATA Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	RM_DATA	Memory Data Register for Read/Write Transactions Controlled by RM_ADDR_CTRL	R/W

4.4.170 RM_STATUS — PRP Memory Controller Status Indication

Address: 4405 3D2Ch

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ageaddress															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.172 RM_STATUS Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b16	Reserved	Set to zero on Write. ignore on Read.	R
b15 to b0	ageaddress	Address the aging process will inspect when the aging timer expires next time.	R

4.4.171 TxSeqTooLate — PRP Frame Transmission Retrieval of Failed Sequence

Address: 4405 3D30h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	TxSeqTooLate				
Value after reset	X	X	X	X	X	X	X	X	X	X	X	0	0	0	0	0

Table 4.173 TxSeqTooLate Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b5	Reserved	Set to zero on Write. ignore on Read.	R
b4 to b0	TxSeqTooLate	Retrieval of a sequence number failed. A frame was transmitted and duplicated without appending the RCT. One bit per port, latched high when such an event occurred. Writing the register with “1” in the corresponding bit position will clear the latch. This can happen on ports that transmit to the redundant ports hence the RedBox function will require to retrieve the sequence number for the frame (based on its source address). If that source address lookup took too long and the frame got forwarded before the lookup could complete, the frame is transmitted without appending an RCT (it is still duplicated). This could happen on Gigabit ports for short frames if the switch system frequency is too low and all ports request address lookups simultaneously. ***Note*)** Latches are cleared also when PRP_CONFIG.PRP_ENA = 0.	R/W

4.4.172 CntErrWrongLanA — PRP Wrong ID LAN-A Count Register

Address: 4405 3D34h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	CntErrWrongLanA															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CntErrWrongLanA															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.174 CntErrWrongLanA Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	CntErrWrongLanA	This field is not available.	R

4.4.173 CntErrWrongLanB — PRP Wrong ID LAN-B Count Register

Address: 4405 3D38h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	CntErrWrongLanB															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CntErrWrongLanB															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.175 CntErrWrongLanB Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	CntErrWrongLanB	This field is not available.	R

4.4.174 CntDupLanA — PRP Duplicate LAN-A Count Register

Address: 4405 3D3Ch

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	CntDupLanA															
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CntDupLanA															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.176 CntDupLanA Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	CntDupLanA	Valid frames received on LAN A that were dropped by duplicate detection	R

4.4.175 CntDupLanB — PRP Duplicate LAN-B Count Register

Address: 4405 3D40h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	CntDupLanB															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CntDupLanB															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.177 CntDupLanB Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	CntDupLanB	Valid frames received on LAN B that were dropped by duplicate detection	R

4.4.176 CntOutOfSeqLowA — PRP Sequence Error Low LAN-A Count Register

Address: 4405 3D44h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	CntOutOfSeqLowA															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CntOutOfSeqLowA															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.178 CntOutOfSeqLowA Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	CntOutOfSeqLowA	Valid and accepted frames received on LAN A with a sequence number less than last - window (DUP_W)	R

4.4.177 CntOutOfSeqLowB — PRP Sequence Error Low LAN-B Count Register

Address: 4405 3D48h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	CntOutOfSeqLowB															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CntOutOfSeqLowB															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.179 CntOutOfSeqLowB Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	CntOutOfSeqLowB	Valid and accepted frames received on LAN B with a sequence number less than last - window (DUP_W)	R

4.4.178 CntOutOfSeqA — PRP Sequence Error LAN-A Count Register

NOTE

Increments also with CntOutOfSeqLowA

Address: 4405 3D4Ch

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	CntOutOfSeqA															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CntOutOfSeqA															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.180 CntOutOfSeqA Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	CntOutOfSeqA	Valid and accepted frames received on LAN A with an unexpected sequence number	R

4.4.179 CntOutOfSeqB — PRP Sequence Error LAN-B Count Register

NOTE

Increments also with CntOutOfSeqLowB

Address: 4405 3D50h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	CntOutOfSeqB															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CntOutOfSeqB															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.181 CntOutOfSeqB Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	CntOutOfSeqB	Valid and accepted frames received on LAN B with an unexpected sequence number	R

4.4.180 CntAcceptA — PRP Valid Frame LAN-A Count Register

Address: 4405 3D54h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	CntAcceptA															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CntAcceptA															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.182 CntAcceptA Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	CntAcceptA	Valid frames received on LAN A which had a valid sequence number in the expected range	R

4.4.181 CntAcceptB — PRP Valid Frame LAN-B Count Register

Address: 4405 3D58h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	CntAcceptB															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CntAcceptB															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.183 CntAcceptB Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	CntAcceptB	Valid frames received on LAN B which had a valid sequence number in the expected range	R

4.4.182 CntMissing — PRP Drop history Adjustment Count

Address: 4405 3D5Ch

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	CntMissing															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CntMissing															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.184 CntMissing Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	CntMissing	Indicates adjustment of the drop history as a frame was received with a sequence number of expected + history + 1. This occurs if the same frame was dropped in both LAN segments (one sequence number is missing) and the history is now extended beyond that sequence number (causing it to be treated as drop allowed).	R

4.4.183 HUB_CONFIG — HUB Configuration Register

Address: 4405 3E00h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	HUB_ISOLATE	TRIGGER_MODE	RETRANSMIT_ENA	HUB_ENA
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	0	0	0	0

Table 4.185 HUB_CONFIG Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b4	Reserved	Set to zero on Write. ignore on Read.	R
b3	HUB_ISOLATE	Isolates all hub ports from the other ports of the switch and allows communication with management port only. It is then up to the management port's application to implement some bridging functionality to other ports as required. When 1, frames received at any of the Hub ports are forwarded to the management port always. No lookup or flooding occurs. In addition, all frames from ports outside the Hub Group that would need forwarding to the Hub Group are also forwarded to the management port always. The setting has no effect on forwarding in between non hub ports and forwarding to/from the management port. When 0, normal forwarding is performed and the HUB_DEFPORT mask is used when forwarding traffic to the hub group from any other port. ***Note*)** This setting only controls forwarding within the switch. The Hub PHY copy function is not affected by this setting.	R/W
b2	TRIGGER_MODE	Enable single frame trigger mode. The HUB will send a single frame (single MAC transmit) only when allowed by a trigger event. Trigger events are generated by the settings in the HUB_TRIGGER_IMMEDIATE and HUB_TRIGGER_AT registers, individually per port. ***Note*)** The RETRANSMIT_ENA must be 0 when trigger mode is active.	R/W
b1	RETRANSMIT_ENA	Enable Hub retransmit capability. When 1 and a collision occurs while a MAC of the Hub Group is transmitting, the MAC is allowed to retransmit if possible (following IEEE 802.3 half-duplex backoff rules). When 0 and a collision occurs, the MAC will not retransmit the frame and the frame is discarded. ***Note*)** Must be 0 when TRIGGER_MODE = 1.	R/W
b0	HUB_ENA	Enable Integrated HUB operation. All other configuration registers must be configured correctly before setting this bit. When cleared, the HUB statistics counters get cleared also.	R/W

4.4.184 HUB_GROUP — HUB Port Group Register

NOTE

Only setting a single port would allow the port to use the trigger capabilities.

Address: 4405 3E04h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	HUB_GROUP			
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	0	0	0	0

Table 4.186 HUB_GROUP Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b4	Reserved	Set to zero on Write. ignore on Read.	R
b3 to b0	HUB_GROUP	Defines all ports that should be combined to a Hub Group. A portmask with at least 1 bit set (typically at least 2 are set). The management port cannot be used (write ignored). The MACs of the Hub group must be configured to identical speed and half-duplex operation. The group should not collide with any other group settings of the switch that may be active (e.g. DLR, PRP) as this can lead to unpredictable behavior.	R/W

4.4.185 HUB_DEFPORT — HUB Default Port Selection Register

NOTE

The management port's forced forwarding can be used to direct frames to other ports within the group.

Address: 4405 3E08h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	HUB_DEFPORT			
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	0	0	0	0

Table 4.187 HUB_DEFPORT Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b4	Reserved	Set to zero on Write. ignore on Read.	R
b3 to b0	HUB_DEFPORT	The default port within the Hub Group where all traffic from a port outside the group is forwarded to. If a frame should be forwarded to any of the hub ports the frame is sent to this port only. The hub's copy function will copy it to all PHY interfaces of the group eventually. Set portmask with exactly 1 bit from the hub group.	R/W

4.4.186 HUB_TRIGGER_IMMEDIATE — HUB Transmission Trigger Immediate Register

NOTE

If no frame is waiting at the port at the time of writing this register, the trigger will be kept active until a frame arrives. It then is transmitted and the register clears.

Address: 4405 3E0Ch

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	HUB_TRIGGER_IMMEDIATE			
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	0	0	0	0

Table 4.188 HUB_TRIGGER_IMMEDIATE Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b4	Reserved	Set to zero on Write. ignore on Read.	R
b3 to b0	HUB_TRIGGER_IMMEDIATE	Trigger Transmission of a Single Frame from Given Port within the Hub Group Portmask with at maximum having a single bit set. Setting a port bit that is not within the hub group has no effect (i.e. register write is ignored). When written, the specified port is allowed to send a single frame (as soon as possible). The register is cleared when transmission has started and an interrupt can be generated.	R/W

4.4.187 HUB_TRIGGER_AT — HUB Transmission Trigger At Register

NOTE

The register clears also if the port has no frame to transmit at the time of the trigger. Then the trigger interrupt will occur and the register clears but no frame is transmitted. A frame arriving after the trigger time will then not be transmitted (this is in contrast to the trigger immediate function).

Address: 4405 3E10h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	HUB_TRIGGER_AT			
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	0	0	0	0

Table 4.189 HUB_TRIGGER_AT Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b4	Reserved	Set to zero on Write. ignore on Read.	R
b3 to b0	HUB_TRIGGER_AT	Trigger Transmission of a Single Frame at a Specific Time. Portmask with at maximum having a single bit set. Setting a port bit that is not within the hub group has no effect (i.e. register write is ignored). The HUB_TTIME register should be written first to avoid unexpected triggering on an older/invalid timer value. The register is cleared when transmission has started and an interrupt can be generated.	R/W

4.4.188 HUB_TTIME — HUB Transmission Time Define Register

Address: 4405 3E14h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	HUB_TTIME															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	HUB_TTIME															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.190 HUB_TTIME Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	HUB_TTIME	Defines the time value when a trigger should occur. A 32-bit value compared with the timer. The value is compared with the hardware timer (input A5PSW_TS_NS_IN[31:0]) and if the timer reaches (or crosses) the value, the port given in HUB_TRIGGER_AT is allowed to transmit one frame.	R/W

4.4.189 HUB_IRQ_CONTROL — HUB Interrupt Control Register

If any of the interrupt enable bits is set and a corresponding event occurs, the Hub interrupt within the global INT_STAT_ACK interrupt status register asserts. To clear any of the interrupts, the HUB_IRQ_STAT_ACK register must be written.

Address: 4405 3E18h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	TRIGGER_TIMER_ACK	TRIGGER_IMMEDIATE_ACK	CHANGE_DET	RX_TRIGGER			
Value after reset	X	X	X	X	X	X	X	X	X	0	0	0	0	0	0	0

Table 4.191 HUB_IRQ_CONTROL Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b7	Reserved	Set to zero on Write. ignore on Read.	R
b6	TRIGGER_TIMER_ACK	Enable interrupt when Hub transmit started after writing the HUB_TRIGGER_AT register and the timeout value has been reached (register HUB_TTIME). The event is caused when the transmission has started and the HUB_TRIGGER_AT register has been cleared.	R/W
b5	TRIGGER_IMMEDIATE_ACK	Enable interrupt when Hub transmit started after writing the HUB_TRIGGER_IMMEDIATE register. The event is caused when the transmission has started and the HUB_TRIGGER_IMMEDIATE register has been cleared and can accept a new command.	R/W
b4	CHANGE_DET	Enable interrupt for Hub TX statemachine port state change request detection. Asserts when the HUB_STATUS register bit TX Change Pending asserts.	R/W
b3 to b0	RX_TRIGGER	Enable interrupt on receive pattern match trigger function. One bit per port. Bit 0 = port 0, Bit 1 = port 1, ..., Bit 3 = port 3. When enabled, an interrupt is generated when the receive pattern matcher on that port caused the Hub transmit to transmit one frame (trigger mode) and transmission has begun. ***Note)*** This interrupt is in addition to the pattern's own match interrupt capability that can be enabled in the PATTERN_CTRL register. This trigger interrupt will occur typically shortly after the pattern match interrupt did occur (e.g. up to 1μs later in a 100 Mbps port which is the time between frame receive and transmit start following the IPG delay). Hence it can cause two separate interrupts for the same event to the system if both are enabled, which may cause unwanted processing overhead.	R/W

4.4.190 HUB_IRQ_STAT_ACK — HUB Interrupt Status/ACK Register

To clear any of the interrupts, the HUB_IRQ_STAT_ACK register must be written.

Address: 4405 3E1Ch

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	TRIGGER_TIMER_ACK	TRIGGER_IMMEDIATE_ACK	CHANGE_DET	RX_TRIGGER			
Value after reset	X	X	X	X	X	X	X	X	X	0	0	0	0	0	0	0

Table 4.192 HUB_IRQ_STAT_ACK Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b7	Reserved	Set to zero on Write. ignore on Read.	R
b6	TRIGGER_TIMER_ACK	Interrupt Pending Indication Bits corresponding to the bits defined in HUB_IRQ_CONTROL. If an event occurs, the corresponding bit is latched high. To clear the bit, it must be written with a “1”. ***Note)*** The latches operate independently of the HUB_IRQ_CONTROL register. That is, even when the interrupt is disabled, the event occurrences can still be inspected.	R/W
b5	TRIGGER_IMMEDIATE_ACK	Interrupt Pending Indication Bits corresponding to the bits defined in HUB_IRQ_CONTROL. If an event occurs, the corresponding bit is latched high. To clear the bit, it must be written with a “1”. ***Note)*** The latches operate independently of the HUB_IRQ_CONTROL register. That is, even when the interrupt is disabled, the event occurrences can still be inspected.	R/W
b4	CHANGE_DET	Interrupt Pending Indication Bits corresponding to the bits defined in HUB_IRQ_CONTROL. If an event occurs, the corresponding bit is latched high. To clear the bit, it must be written with a “1”. ***Note)*** The latches operate independently of the HUB_IRQ_CONTROL register. That is, even when the interrupt is disabled, the event occurrences can still be inspected.	R/W
b3 to b0	RX_TRIGGER	Interrupt Pending Indication Bits corresponding to the bits defined in HUB_IRQ_CONTROL. If an event occurs, the corresponding bit is latched high. To clear the bit, it must be written with a “1”. ***Note)*** The latches operate independently of the HUB_IRQ_CONTROL register. That is, even when the interrupt is disabled, the event occurrences can still be inspected.	R/W

4.4.191 HUB_STATUS — HUB Status Register

Address: 4405 3E20h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	TX_Change_Pending	Speed_OK	TX_BUSY	TX_ACTIVE	—	—	—	—	—	PORTS_ACTIVE			
Value after reset	X	X	X	0	0	0	0	X	X	X	X	X	0	0	0	0

Table 4.193 HUB_STATUS Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b13	Reserved	Set to zero on Write. ignore on Read.	R
b12	TX_Change_Pending	Indicates a pending change request in the hub transmitter that is unsolved and causes the Hub to stop operation (no longer performing any transmissions). A state change is requested when a port of the hub group changed its operational mode unexpectedly (link fail, MAC speed change). It also will assert if e.g. the Hub is disabled in the middle of a transmission until the transmission has ended normally, which will then allow the hub to become disabled (and eventually deassert TX ACTIVE). The bit will deassert if the misconfiguration has been resolved and the Hub is allowed to perform transmissions. The hub is inoperable if this bit is set and the misconfiguration may need the intervention of the application to be resolved (e.g. speed mismatch persists).	R
b11	Speed_OK	Indicates that the port speed of all group ports match. This bit must be 1 to allow the TX statemachine to enter active state and may indicate a misconfiguration or link failure that changed the speed of the attached PHY of a port. It is important that all PHYs at all time when they are part of the Hub group operate with the same MAC interface speed even if the link would be disconnected (e.g. the PHY must not change its MAC interface speed from 100 Mbps to 10 Mbps after the link dropped). The Hub is inoperable (not transmitting) and cannot enter the active state as long as this bit is 0. The bit is valid as soon as the hub group is defined (HUB_GROUP) and enabled (HUB_CONFIG). It may give an indication why the TX ACTIVE is not getting set.	R
b10	TX_BUSY	The local device currently transmits data to all ports within the hub group. The bit is informal and will toggle during normal operation.	R
b9	TX_ACTIVE	The hub global transmit statemachine has successfully entered Hub mode and is now controlling the hub group. The bit will react with some delay following a write into the HUB_CONFIG/HUB_GROUP registers until it successfully could enter or leave the hub operational mode. The statemachine may not enter hub mode even when enabled through the HUB_CONFIG register, as long as the ports are still transferring data or a misconfiguration exists (e.g. speed differences, no group defined).	R
b8 to b4	Reserved	Set to zero on Write. ignore on Read.	R
b3 to b0	PORTS_ACTIVE	Shows the currently active ports of the Hub group which are allowed for transmit. It can differ from the HUB_GROUP setting if a port is e.g. disabled, or has no link. One bit per port. Bit 0 = Port 0, Bit 1 = Port 1, …	R

4.4.192 HUB_OPORT_STATUS — HUB Output Port Status Register

Address: 4405 3E24h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	HUB_OPORT_STATUS			
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	0	0	0	0

Table 4.194 HUB_OPORT_STATUS Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b4	Reserved	Set to zero on Write. ignore on Read.	R
b3 to b0	HUB_OPORT_STATUS	Per Output Port Data Available Status One bit per port. A bit = 1 indicates the output port has data available for transmission (any queue of the port has data). This is a real time indication of all line ports of the switch, not only the ones for the hub.	R

4.4.193 RXMATCH_CONFIG[n] — RX Pattern Match Configuration for PORT[n] (n = 0..4)

Address: 4405 3E80h + 4h × n

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	PATTERN_EN							
Value after reset	X	X	X	X	X	X	X	X	0	0	0	0	0	0	0	0

Table 4.195 RXMATCH_CONFIG[n] Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b8	Reserved	Set to zero on Write. ignore on Read.	R
b7 to b0	PATTERN_EN	Enables pattern(s) on the port (RX). One bit per pattern. When a bit x is set, the corresponding pattern is searched within frames received at the port. The pattern's match function executed is defined in its PATTERN_CTRL registers. Multiple patterns can be enabled at the same time (allowing any match). ***Note)*** If multiple patterns are enabled, they should be exclusive to each other. If multiple patterns match simultaneously, the executing function is unpredictable if the controls differ.	R/W

4.4.194 PATTERN_CTRL[n] — RX Pattern Match Function Control for Pattern[n] (n = 0..7)

For every pattern data set, one pattern control register defines the function that is executed when the pattern matches onto a received frame. Whether a pattern is applied to received traffic of a port is controlled by the RXMATCH_CONFIG registers.

Note that any control bit acts independent of each other. Hence if multiple bits are set, those functions will all execute and/or affect the frame simultaneously.

Address: 4405 3EB0h + 4h × n

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	PORTMASK				
Value after reset	X	X	X	X	X	X	X	X	X	X	X	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	PRIORITY		—	—	—	HUBTRIGGER	—	—	—	—	SET_PRIO	DISCARD	MGMTFWD	MATCH_NOT
Value after reset	X	X	0	0	X	X	X	0	X	X	X	X	0	0	0	0

Table 4.196 PATTERN_CTRL[n] Register Contents (1/2)

Bit Position	Bit Name	Function	R/W
b31 to b21	Reserved	Set to zero on Write. ignore on Read.	R
b20 to b16	PORTMASK	A Port Mask Used Depending on the Control Bits below (e.g. HUBTRIGGER) One bit per port: Bit 16 = port 0, Bit 17 = port 1, …	R/W
b15, b14	Reserved	Set to zero on Write. ignore on Read.	R
b13, b12	PRIORITY	Priority of the frame used when SET_PRIO is set. The priority is used to forward the frame into the corresponding output queue of a port. A higher value defines a higher priority.	R/W
b11 to b9	Reserved	Set to zero on Write. ignore on Read.	R
b8	HUBTRIGGER	When set, the port defined in the PORTMASK setting (see above) is allowed for transmitting one frame. Usable for Hub mode only and if the Hub has been configured to operate in Trigger mode. ***Note)*** The port mask should contain only a single port from within the hub group. A hub trigger is a global event and can occur only once until the enabled port transmits the frame. Hence, if multiple patterns would match simultaneously, only one trigger (which one is unknown) will execute. All other triggers that may follow will be ignored as long as a pending trigger has not been processed (i.e. frame transmit has not started yet).	R/W
b7 to b4	Reserved	Set to zero on Write. ignore on Read.	R
b3	SET_PRIO	Set frame priority, overriding normal classification. When a match occurs, the frame's priority is set as given in the PRIORITY bits (see above). This priority takes precedence over (i.e. ignores) settings in the PRIORITY_CFG[n] register of the port where the frame is received.	R/W
b2	DISCARD	When set, the frame is discarded.	R/W
b1	MGMTFWD	When set, the frame is forwarded to the management port only (suppressing destination address lookup). ***Note)*** If the pattern is applied to the management port, this setting has no effect.	R/W

Table 4.196 PATTERN_CTRL[n] Register Contents (2/2)

Bit Position	Bit Name	Function	R/W
b0	MATCH_NOT	When set, a match is reported and the functions of this control are executed if the pattern does not match. When cleared (default), the above control executes when the pattern matches on an incoming frame. This bit is available only in PATTERN_CTRL0. It is ignored and must be 0 in all other pattern control registers. When set, the PATTERN_CTRL0 will be used if no patterns (including PATTERN_CTRL0) report a match and the PATTERN_CTRL0 has been enabled for a port.	R/W

4.4.195 PTN_IRQ_CONTROL — RX Pattern Match Interrupt Control Register

If any of the patterns match (globally) and a pattern's interrupt enable bits is set, the PATTERN interrupt within the global INT_STAT_ACK interrupt status register asserts. To clear any of the interrupts, the PTN_IRQ_STAT_ACK register must be written.

Address: 4405 3ED0h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	MATCHINT							
Value after reset	X	X	X	X	X	X	X	X	0	0	0	0	0	0	0	0

Table 4.197 PTN_IRQ_CONTROL Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b8	Reserved	Set to zero on Write. ignore on Read.	R
b7 to b0	MATCHINT	Enable Interrupt on Receive Pattern Match One bit per pattern. Bit 0 = pattern 0, ..., Bit 7 = pattern 7. If set and the corresponding pattern match occurs when processing received frames (on any port), an interrupt occurs. The corresponding interrupt signal (A5PSW_PTRN_Int) will also assert as long as the interrupt is not cleared. ***Note)*** The interrupt occurs following the CRC check of the frame by the MAC RX. Even if the frame would be dropped by the switch eventually due to other rules, the match interrupt can still occur (it will not occur if the frame is dropped by the MAC).	R/W

4.4.196 PTN_IRQ_STAT_ACK — RX Pattern Match Interrupt Status/ACK Register

Address: 4405 3ED4h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	MATCHINT							
Value after reset	X	X	X	X	X	X	X	X	0	0	0	0	0	0	0	0

Table 4.198 PTN_IRQ_STAT_ACK Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b8	Reserved	Set to zero on Write. ignore on Read.	R
b7 to b0	MATCHINT	Interrupt pending indication for the corresponding pattern match events (see PTN_IRQ_CONTROL). If an event occurs, the corresponding bit is latched high. To clear the bit, it must be written with a “1”. ***Note)*** The latches operate independently of the PTN_IRQ_CONTROL register. That is, even when the interrupt is disabled, the event occurrences can still be inspected.	R/W

4.4.197 PATTERN_SEL — RX Pattern Number Selection Register

Address: 4405 3EDCh

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	PATTERN_SEL		
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	0	0	0

Table 4.199 PATTERN_SEL Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b3	Reserved	Set to zero on Write. ignore on Read.	R
b2 to b0	PATTERN_SEL	Defines the pattern number which is selected for read/write through the following registers PTRN_CMP_*, PTRN_MSK_*. A read or write operation into any of the PTRN_CMP and PTRN_MSK registers will affect the pattern that is selected by this register. To configure a specific pattern dataset, first this selection register must be set to the pattern number of interest.	R/W

4.4.198 PTRN_CMP_30 — Pattern Compare Value Bytes 3 .. 0

Address: 4405 3EE0h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	PTRN_CMP_30															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PTRN_CMP_30															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.200 PTRN_CMP_30 Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	PTRN_CMP_30	Pattern Compare Value Bytes 3 .. 0 First byte (Byte 0) is 7:0. 4th byte (Byte 3) is 31:24.	R/W

4.4.199 PTRN_CMP_74 — Pattern Compare Value Bytes 7 .. 4

Address: 4405 3EE4h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	PTRN_CMP_74															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PTRN_CMP_74															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.201 PTRN_CMP_74 Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	PTRN_CMP_74	Pattern Compare Value Bytes 7 .. 4 First byte (Byte 4) is 7:0. 4th byte (Byte 7) is 31:24.	R/W

4.4.200 PTRN_CMP_118 — Pattern Compare Value Bytes 11 .. 8

Address: 4405 3EE8h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	PTRN_CMP_118															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PTRN_CMP_118															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.202 PTRN_CMP_118 Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	PTRN_CMP_118	Pattern Compare Value Bytes 11 .. 8 First byte (Byte 8) is 7:0. 4th byte (Byte 11) is 31:24.	R/W

4.4.201 PTRN_MSK_30 — Pattern Mask for Bytes 3 .. 0

Address: 4405 3EF0h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	PTRN_MSK_30															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PTRN_MSK_30															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.203 PTRN_MSK_30 Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	PTRN_MSK_30	Pattern Mask for Bytes 3 .. 0	R/W

4.4.202 PTRN_MSK_74 — Pattern Mask for Bytes 7 .. 4

Address: 4405 3EF4h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	PTRN_MSK_74															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PTRN_MSK_74															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.204 PTRN_MSK_74 Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	PTRN_MSK_74	Pattern Mask for Bytes 7 .. 4	R/W

4.4.203 PTRN_MSK_118 — Pattern Mask for Bytes 11 .. 8

Address: 4405 3EF8h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	PTRN_MSK_118															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PTRN_MSK_118															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.205 PTRN_MSK_118 Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	PTRN_MSK_118	Pattern Mask for Bytes 11 .. 8	R/W

4.4.204 TDMA_CONFIG — TDMA Configuration Register

Address: 4405 3F00h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	WAIT_START	TDMA_ENA
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	0	0

Table 4.206 TDMA_CONFIG Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b2	Reserved	Set to zero on Write. ignore on Read.	R
b1	WAIT_START	Status bit which is set as long as the scheduler is enabled but has not yet reached the time given in register TDMA_START. The bit clears when the start time has been reached and the scheduler has begun with the very first cycle.	R
b0	TDMA_ENA	Enable TDMA Scheduler The bit should be set only after all other TDMA scheduler configuration registers have been correctly configured. When set, the scheduler will begin with its first cycle at the time given in TDMA_START. Before that time is reached, the queues defined in QUEUES_START will be enabled Clearing the bit is allowed any time to disable the TDMA scheduler allowing all ports to transmit normally again. ***Caution)*** The scheduler must be enabled before the timer has reached the value in TDMA_START. Enabling it too late would cause the scheduler to start after the timer has wrapped around and then reached the start value.	R/W

4.4.205 TDMA_PORTS — TDMA Scheduling Enable Register

Address: 4405 3F04h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	port4	port3	port2	port1	port0
Value after reset	X	X	X	X	X	X	X	X	X	X	X	0	0	0	0	0

Table 4.207 TDMA_PORTS Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b5	Reserved	Set to zero on Write. ignore on Read.	R
b4	port4	Port4 Define the ports that should use TDMA scheduling.	R/W
b3	port3	Port3 Define the ports that should use TDMA scheduling.	R/W
b2	port2	Port2 Define the ports that should use TDMA scheduling.	R/W
b1	port1	Port1 Define the ports that should use TDMA scheduling.	R/W
b0	port0	Port0 Define the ports that should use TDMA scheduling.	R/W

4.4.206 TDMA_START — TDMA Start Time Set Register

Address: 4405 3F08h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TDMA_START															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TDMA_START															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.208 TDMA_START Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	TDMA_START	Sets the start time for the very first cycle after system initialization has completed. The value will be compared with the system time and when it is reached (crossed), the scheduler begins with its first cycle. The 2nd cycle then will be at TDMA_START + TDMA_CYCLE.	R/W

4.4.207 TDMA_MODULO — TDMA System Timer Modulo

NOTE

It is not expected that the values at A5PSW_TS_NS_IN(31:0) increment continuously. Relevant for this setting is only the possible range of values.

Address: 4405 3F0Ch

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TDMA_MODULO															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TDMA_MODULO															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.209 TDMA_MODULO Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	TDMA_MODULO	The System Timer Modulo It gives the maximum time value+1 the timer input (A5PSW_TS_NS_IN[31:0]) will reach before it wraps around. Setting a value of 0 would indicate a wrap-around at 2^{32}.	R/W

4.4.208 TDMA_CYCLE — TDMA Periodic Cycle Set Register

The cycle time can be manipulated any time during TDMA operation to possibly compensate for a drifting system timer (however usually the system timer should be adjusted and locked to a master clock).

Address: 4405 3F10h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TDMA_CYCLE															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TDMA_CYCLE															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.210 TDMA_CYCLE Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	TDMA_CYCLE	The Periodic Cycle Time for the Scheduler Given in system timer time.	R/W

4.4.209 TDMA_T1 — TDMA 1st Time Offset

NOTE

Must be greater than 10 timer value steps (i.e. greater than increment within 10 timer module clock cycles).

Address: 4405 3F14h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TDMA_T1															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TDMA_T1															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.211 TDMA_T1 Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	TDMA_T1	First Time Offset from Cycle Start −1 Given in system timer time. Defines the end of the first slot within every cycle.	R/W

4.4.210 TDMA_T2 — TDMA 2nd Time Offset

NOTE

Must be greater than T1 by at least 10 timer value steps (i.e. greater than increment within 10 timer module clock cycles).

If T2 and T3 is not needed, T2 can be set to 0 (i.e. if T2 = 0 then T3 is ignored).

Address: 4405 3F18h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TDMA_T2															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TDMA_T2															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.212 TDMA_T2 Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	TDMA_T2	2nd Time Offset from Cycle Start −1 Given in system timer time. Defines the end of the 2nd slot.	R/W

4.4.211 TDMA_T3 — TDMA 3rd Time Offset

NOTE

Must be greater than T2 by at least 10 timer value steps (i.e. greater than increment within 10 timer module clock cycles).

It must be smaller than the cycle time with a maximum less than the same distance of 10 timer steps to the cycle time.

If T3 is not needed, it can be set to 0.

Address: 4405 3F1Ch

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TDMA_T3															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TDMA_T3															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.213 TDMA_T3 Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	TDMA_T3	3rd Time Offset from Cycle Start −1 Given in system timer time. Defines the end of the 3rd slot.	R/W

4.4.212 QUEUES_TS — TDMA 1st Slot Transmit Enable

Multiple bits as well as none can be set. When multiple bits are set, priority selection applies between the enabled queues and the frames from the highest queue will be sent first. If no bit is set, no queue will transmit during the slot.

NOTE

The setting applies to all ports that are configured for TDMA mode in register TDMA_PORTS.

Address: 4405 3F20h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	queue3	queue2	queue1	queue0
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	0	0	0	0

Table 4.214 QUEUES_TS Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b4	Reserved	Set to zero on Write. ignore on Read.	R
b3	queue3	Queue3 Allowed for Transmit at Cycle Start (until T1)	R/W
b2	queue2	Queue2 Allowed for Transmit at Cycle Start (until T1)	R/W
b1	queue1	Queue1 Allowed for Transmit at Cycle Start (until T1)	R/W
b0	queue0	Queue0 Allowed for Transmit at Cycle Start (until T1)	R/W

4.4.213 QUEUES_T1 — TDMA 2nd Slot Transmit Enable

Address: 4405 3F24h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	queue3	queue2	queue1	queue0
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	0	0	0	0

Table 4.215 QUEUES_T1 Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b4	Reserved	Set to zero on Write. ignore on Read.	R
b3	queue3	Queue3 Allowed to Transmit from Time T1 until T2 (2nd slot)	R/W
b2	queue2	Queue2 Allowed to Transmit from Time T1 until T2 (2nd slot)	R/W
b1	queue1	Queue1 Allowed to Transmit from Time T1 until T2 (2nd slot)	R/W
b0	queue0	Queue0 Allowed to Transmit from Time T1 until T2 (2nd slot)	R/W

4.4.214 QUEUES_T2 — TDMA 3rd Slot Transmit Enable

Address: 4405 3F28h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	queue3	queue2	queue1	queue0
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	0	0	0	0

Table 4.216 QUEUES_T2 Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b4	Reserved	Set to zero on Write. ignore on Read.	R
b3	queue3	Queue3 Allowed to Transmit from Time T2 until T3 (3rd slot)	R/W
b2	queue2	Queue2 Allowed to Transmit from Time T2 until T3 (3rd slot)	R/W
b1	queue1	Queue1 Allowed to Transmit from Time T2 until T3 (3rd slot)	R/W
b0	queue0	Queue0 Allowed to Transmit from Time T2 until T3 (3rd slot)	R/W

4.4.215 QUEUES_T3 — TDMA Last Slot Transmit Enable

Address: 4405 3F2Ch

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	queue3	queue2	queue1	queue0
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	0	0	0	0

Table 4.217 QUEUES_T3 Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b4	Reserved	Set to zero on Write. ignore on Read.	R
b3	queue3	Queue3 Allowed to Transmit from Time T3 until End of Cycle (last slot before next cycle starts)	R/W
b2	queue2	Queue2 Allowed to Transmit from Time T3 until End of Cycle (last slot before next cycle starts)	R/W
b1	queue1	Queue1 Allowed to Transmit from Time T3 until End of Cycle (last slot before next cycle starts)	R/W
b0	queue0	Queue0 Allowed to Transmit from Time T3 until End of Cycle (last slot before next cycle starts)	R/W

4.4.216 QUEUES_START — TDMA First Cycle Transmit Enable

Address: 4405 3F30h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	queue3	queue2	queue1	queue0
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	0	0	0	0

Table 4.218 QUEUES_START Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b4	Reserved	Set to zero on Write. ignore on Read.	R
b3	queue3	Queue3 allowed to transmit when the scheduler was enabled until the system timer has reached the first cycle time given in TDMA_START.	R/W
b2	queue2	Queue2 allowed to transmit when the scheduler was enabled until the system timer has reached the first cycle time given in TDMA_START.	R/W
b1	queue1	Queue1 allowed to transmit when the scheduler was enabled until the system timer has reached the first cycle time given in TDMA_START.	R/W
b0	queue0	Queue0 allowed to transmit when the scheduler was enabled until the system timer has reached the first cycle time given in TDMA_START.	R/W

4.4.217 TIME_LOAD_NEXT — TDMA Calculated Next Loading Time

Address: 4405 3F34h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TIME_LOAD_NEXT															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TIME_LOAD_NEXT															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.219 TIME_LOAD_NEXT Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	TIME_LOAD_NEXT	Status giving the calculated time the scheduler will load into its internal compare register after the current running slot end will be reached. (i.e. it is not the end of the current slot).	R

4.4.218 TDMA_IRQ_CONTROL — TDMA Interrupt Control Register

Address: 4405 3F38h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	T3_EN	T2_EN	T1_EN	CYCLE_EN
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	0	0	0	0

Table 4.220 TDMA_IRQ_CONTROL Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b4	Reserved	Set to zero on Write. ignore on Read.	R
b3	T3_EN	Enable interrupt on TDMA cycle T3 offset reached. The interrupt occurs every cycle.	R/W
b2	T2_EN	Enable interrupt on TDMA cycle T2 offset reached. The interrupt occurs every cycle.	R/W
b1	T1_EN	Enable interrupt on TDMA cycle T1 offset reached. The interrupt occurs every cycle.	R/W
b0	CYCLE_EN	Enable interrupt on TDMA cycle start. The interrupt occurs every cycle. ***Note)*** Asserts also when the very first cycle started after enabling the scheduler (i.e. at time TDMA_START)	R/W

4.4.219 TDMA_IRQ_STAT_ACK — TDMA Interrupt Status/ACK Register

Address: 4405 3F3Ch

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	T3_EN	T2_EN	T1_EN	CYCLE_EN
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	0	0	0	0

Table 4.221 TDMA_IRQ_STAT_ACK Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b4	Reserved	Set to zero on Write. ignore on Read.	R
b3	T3_EN	Interrupt pending indication for the corresponding events.*1 If an TDMA cycle T3 offset reached, the corresponding bit is latched high. To clear the bit, it must be written with a “1”. ***Note)*** The latches operate independently of the TDMA_IRQ_CONTROL register. That is, even when the interrupt is disabled, the event occurrences can still be inspected.	R/W
b2	T2_EN	Interrupt pending indication for the corresponding events.*1 If an TDMA cycle T2 offset reached, the corresponding bit is latched high. To clear the bit, it must be written with a “1”. ***Note)*** The latches operate independently of the TDMA_IRQ_CONTROL register. That is, even when the interrupt is disabled, the event occurrences can still be inspected.	R/W
b1	T1_EN	Interrupt pending indication for the corresponding events.*1 If an TDMA cycle T1 offset reached, the corresponding bit is latched high. To clear the bit, it must be written with a “1”. ***Note)*** The latches operate independently of the TDMA_IRQ_CONTROL register. That is, even when the interrupt is disabled, the event occurrences can still be inspected.	R/W
b0	CYCLE_EN	Enable interrupt on TDMA cycle start. The interrupt occurs every cycle. ***Note)*** Asserts also when the very first cycle started after enabling the scheduler (i.e. at time TDMA_START)	R/W

Note 1. See **Section 4.4.218, TDMA_IRQ_CONTROL — TDMA Interrupt Control Register**.

4.5 Operation

4.5.1 Ethernet Frame Format Overview

4.5.1.1 Overview

The IEEE 802.3 Standard defines the Ethernet Frame format as follows: An Ethernet frame has a minimum length of 64 bytes and a maximum length of 1518 bytes, excluding the preamble and the SFD bytes. An Ethernet frame consists of the following fields:

- Seven bytes preamble
- Start frame delimiter (SFD)
- Two address fields
- Length or type field
- Data field
- Frame check sequence (CRC value)
- An EXTENSION field is defined only for half-duplex implementations and is not supported

Figure 4.2 MAC Frame Format Overview

Optionally MAC frames can be VLAN tagged with an additional 4 Byte field (TPID and VLAN Info) inserted between the MAC Source Address and the Type/Length Field. VLAN tagging is defined by the IEEE P802.1Q specification. VLAN tagged frames have a maximum length of 1522 bytes (Tagged standard frames) or 9022 bytes (Tagged Jumbo frames), excluding the preamble and the SFD bytes.

Figure 4.3 VLAN Tagged MAC Frame Format Overview

Table 4.222 MAC Frame Definition

Term	Description
Frame length	The length, in octets, defines the length of the complete Frame without preamble and SFD. A frame has a valid length if it contains at least 64 octets and does not exceed the programmed maximum length (typical 1518).
Payload Length	The length / type field indicates the length of the frame's payload section or identifies the type of the frame. The most significant byte is sent/received first. ● If the Length/type field is set to a value less 1536 (0x600), it is interpreted as a length field indicating number of payload octets following. ***Note)*** A Length/type field less 46 indicates the payload is padded so that the minimum frame length requirement (64 Bytes) is met. For VLAN tagged frames, a value less than 42 indicates a padded frame. ● If the length / type field is set to a value larger or equal to 1536 (0x600), it is interpreted as a type field.
Destination and Source Address	48-Bit MAC addresses. The least significant byte is sent/received first and the two first bits (Two Least Significant bits) of the MAC address are used to distinguish MAC frames.

In typical Ethernet switching application, the MAC, on receive, should be programmed to accept every frame (Promiscuous mode), to check but not change the frame FCS and forward the frame with the FCS field to the Switch. On transmit, the MAC should be programmed not to overwrite the source MAC address received from the Switch and to transmit the frame with the FCS received from the switch.

However when frame manipulation functions within the switch are used, CRC should be stripped by the MAC receivers and will be appended by the MAC transmitters.

4.5.1.2 MAC Address Overview

The destination address bit 0 is used to differentiate Multicast and Unicast Addresses:

- If bit 0 is set “0”, the MAC address is an individual (Unicast) address.
- If bit 0 is set “1”, the MAC address defines a group address (Multicast Address).
- If all 48 bits of the MAC address are set to “1”, it indicates a broadcast address.

In addition, the first 24 Bit of the MAC address define a vendor ID.

Figure 4.4 MAC Address Overview

4.5.1.3 VLAN Tag Overview

A VLAN tagged frame provides a 16-bit VLAN information field. For VLAN priority resolution (see **Section 4.5.3.5(2), VLAN Priority Look Up**), the switch uses the uppest 4 bits of the first word (octet) of the VLAN Info field, that is the 3 bits priority field and the 1 bit CFI field can be used for priority classification.

Figure 4.5 VLAN Info Fields

4.5.1.4 Pause Frames

The receiving device to indicate congestion to the emitting device, which should then stop sending data, generates a Pause Frame.

The Length / Type set to 0x8808 indicates a pause Frame. The two first bytes of a Pause Frame following the type, defines a 16 Bit opcode field set to 0x0001 always. A 16 Bit Pause Quanta is defined in the Frame payload Bytes 2 (Byte P1) and 3 (Byte P2) as defined in the following table. The pause quanta byte P1 is the most significant.

Table 4.223 Pause Frame Format (values in hex)

1	2	3	4	5	6	7	8	9	10	11	12	13	14
55	55	55	55	55	55	55	D5	01	80	C2	00	00	01
Preamble							SFD	Multicast Destination Address					

15	16	17	18	19	20	21	22	23	24	25	26	27 - 68
00	00	00	00	00	00	88	08	00	01	hi	lo	00
Source Address						Type		Opcode		P1	P2	pad (42)

69	70	71	72
xx	xx	xx	xx
CRC-32			

There is no Payload Length field found within a Pause Frame and a Pause Frame is always padded with 42 bytes (0x00). If a pause frame with a pause value greater zero (XOFF Condition) is received, the MAC stops transmitting data as soon the current Frame transfer is completed. The MAC stops transmitting data for the value defined in pause quanta. One pause quanta fraction refers to 512-bit times.

Pause Frames are terminated in the MAC layer and not forwarded by the switch.

4.5.2 IP Frame Format

4.5.2.1 Definitions

The following chapters use the term datagram to describe the protocol specific data unit, which is found within the payload section of its container entity.

For example, an IP datagram specifies the payload section of an Ethernet frame. A TCP datagram specifies the payload section within an IP datagram.

4.5.2.2 Ethernet Types

IP datagrams are carried in the payload section of an Ethernet frame. The Ethernet type/length field is used to discriminate several datagram types. The following table lists the types of interest:

Table 4.224 Ethernet Type Value Examples

Type	Description
8100h	VLAN tagged frame. The actual type is found 4 octets later in the frame
0800h	IPv4
86DDh	IPv6
8808h	MAC Control Frames (Slow Protocols, Flow Control)

4.5.2.3 IPv4 Datagram Format

The following figure shows the IP Version 4 (IPv4) header, which is located at the beginning of an IP datagram. It is organized in 32-bit words. The most significant bit is bit 31 in the following figure. The first byte sent/received is the leftmost byte of the first word (i.e. Version/IHL field).

The IP Header can contain further options, which are always padded if necessary to guarantee the payload following the header is aligned to a 32-bit boundary.

The IP header is followed by the payload immediately, which can contain further protocol headers like e.g. TCP or UDP as indicated by the protocol field value. The complete IP datagram is transported in the payload section of an Ethernet frame.

Figure 4.6 IPv4 Header Format

Table 4.225 IPv4 Header Fields

Field Name	Description
Vers	4-bit IP version information. It is 4 for IPv4 frames.
IHL	4-bit IP header length information. Determines number of 32-bit words found within the IP header. The default value, if no options are present is 5.
TOS	Type of Service / DiffServ field
Length	Total length of the datagram in bytes. It includes all octets of header and payload.
Fragment ID, flags, fragment offset	Fields used for IP fragmentation
TTL	Time-to-live. If zero, datagram must be discarded.
Protocol	Protocol Identifier of protocol that follows in the datagram
Header checksum	Checksum over all IP header fields
Source address	Source IP address
Destination address	Destination IP address

4.5.2.4 IPv6 Datagram Format

The following figure shows the IP Version 6 (IPv6) header, which is located at the beginning of an IP datagram. It is organized in 32-bit words and has a fixed length of 10 words (40 bytes).

The next header field identifies the type of the header to follow the IPv6 header. It is defined identical to the protocol identifier within IPv4 with new definitions for identifying so-called extension headers, which can be inserted between the IPv6 header and the protocol header, shifting the protocol header accordingly.

The most significant bit is bit 31 in the following figure. The first byte sent/received is the leftmost byte of the first word (i.e. Version/Traffic class fields).

Figure 4.7 IPv6 Header Format

Table 4.226 IPv6 Header Fields

Field Name	Description
Vers	4-bit IP version information. It is 6 for all IPv6 frames.
Traffic Class	8-bit field defining the traffic class.
Flow Label	20-bit flow label identifying frames of the same flow.
Payload Length	16-Bit Length of the datagram payload in bytes. It includes all octets following the IPv6 header.
Next Header	Identifies the header that follows the IPv6 header. This can be the protocol header or any IPv6 defined extension header.
Hop limit	Hop counter, decremented by 1 by each station that forwards the frame. If hop limit is 0, the frame must be discarded.
Source Address	128-bit IPv6 source address
Destination Address	128-bit IPv6 destination address

4.5.3 Switch Functional Description

4.5.3.1 Overview

The Switch implements the following main functions:

- Input/Output VLAN Processing
- Input Frame Parsing and Priority Extraction
- Input Port Selection
- Output Port(s) Resolution
- Frame Queuing
- Output Queue Scheduling

The Firmware together with the switching hardware provides a complete Ethernet switching solution.

4.5.3.2 VLAN Input Processing Function

(1) Overview

The VLAN input processing function is used on each switch input port to inspect and manipulate the VLAN tag of frames entering the switch. It performs the following functions:

- Input Frame Parsing
- VLAN tag insertion or manipulation

Based on the information of the Input Processing function, the frame can be switched to the corresponding output port or will be discarded.

VLAN input processing is not performed for BPDU frames.

(2) Terms and Definitions

- VLAN Information: The 16-bit field following the VLAN type field within a frame.
- VLAN ID: The lower 12 bits of the VLAN information field.
- VLAN Priority: The upper 3 bits of the VLAN information field. Used to prioritize incoming frames. A value of 0 represents lowest priority, a value of 7 represents highest priority.

(3) Configuration Information

The switch management provides the following information to configure and control the operation of the function:

- SYSTEM_TAGINFO[n]: 16-bit value. The VLAN information field (VLAN ID and priority) used for tag insertion operations.
- Mode of operation: There are different modes of operation, which define how incoming frames must be processed for a port. The function can be enabled and configured individually per port: See registers VLAN_IN_MODE*[1] and VLAN_IN_MODE_ENA*[2].

Note 1. See **Section 4.4.11, VLAN_IN_MODE — VLAN Input Manipulation Mode Register**.

Note 2. See **Section 4.4.13, VLAN_IN_MODE_ENA — VLAN Input Mode Enable Register**.

CAUTION

If the VLAN input processing function is not enabled (VLAN_IN_MODE_ENA bit of the port = 0), the mode setting has no effect.

(4) Modes of Operation

The VLAN input processing function modifies the frames before they enter the switching engine. This means, if a VLAN tag is inserted, the switch will only act on the inserted VLAN tag (e.g. priority) and any original tag that was found in the frame before the modification, if any, has no effect within the switch.

In addition, if VLAN verification is enabled for a port (see **Section 4.4.5, VLAN_VERIFY — Verify VLAN Domain**), the VLAN id used for insertion (SYSTEM_TAGINFO[n]) must also be configured in the global VLAN resolution table (see **Section 4.4.51, VLAN_RES_TABLE[n] — 32 VLAN Domain Entries (n = 0..31)**), to ensure the switch accepts frames, which contain the inserted tag.

When, in any of the modes, a tag is inserted, it is always inserted as first tag (outer) and its information field is set as programmed in the SYSTEM_TAGINFO[n] register for the port[n] where the frame is received.

Input manipulation can be enabled per port with register VLAN_IN_MODE_ENA and its port individual mode is configured in register VLAN_IN_MODE.

Table 4.227 Input Manipulation Modes

Mode	VLAN_IN_MODE bits [1:0]	Description
1	00b	Single Tagging with Passthrough/VID Overwrite Insert Tag if untagged frame. Leave frame unmodified if tagged and VID > 0. If tagged with VID = 0 (priority tagged), then the VID will be overwritten with the VID from SYSTEM_TAGINFO and priority is kept.
2	01b	Single Tagging with Replace If untagged, add the tag, if single tagged, overwrite the tag.
3	10b	Tag always Insert a tag always. This results in a single tagged frame when an untagged is received, and a double tagged frame, when a single tagged frame is received (or triple tagged if double-tagged received etc.).
4	11b	Reserved mode is not implemented and should not be configured.

4.5.3.3 VLAN Output Processing Function

(1) Overview

The VLAN output processing function is used on a switch output port to manipulate the VLAN tag of the outgoing frames that leave the switch. Frames are processed based on the output Processing mode, and the number of Tags the frame contains.

VLAN output processing is not performed on BPDU frames.

(2) Configuration Information

The switch management provides the information on operating mode to configure and control the operation of the function. There are three different modes of operation, which define how the outgoing frames should be processed.

(3) Modes of Operation

The VLAN output processing function is configured to operate in one of the following modes, which define the way outgoing frames should be treated.

(a) Mode 0: Disabled

No frame manipulation occurs.

(b) Mode 1: Strip Mode

In Strip mode, all the Tags (Single or double) are removed from frame

(c) Mode 2: Tag Through Mode

Always removes first tag from frame only. In Tag Through mode, the inner Tag is passed through while the outer Tag is removed for a double Tagged frame. The following rules apply:

- When a single Tagged frame is received, strip the tag from the frame.
- When a double Tagged frame is received, strip the outer tag from the frame.

(d) Mode 3: VLAN Domain Mode / Transparent Mode

The following function is implemented: VLAN Domain Mode: The first tag of a frame is removed (same as Mode 2) when the VLAN is defined as untagged for the port. The following rules apply:

- If frame's VLAN id is found in the VLAN table (see **Section 4.5.3.9(3)(b), VLAN Domain Resolution / VLAN Table**) and the port is defined as tagged for the VLAN, the frame is not modified.
- If frame's VLAN id is found in the VLAN table and the port is defined as untagged for the VLAN, the first VLAN tag is removed from the frame.
- If frame's VLAN id is not found in the VLAN table, the frame is not modified.

CAUTION

The VLAN table is extended by a 2nd port mask to store the tagged bit for every entry for every port (in addition to the member port mask). This tagged bit mask is accessed using the control bits 30..28 when writing the VLAN_RES_TABLE registers.

4.5.3.4 Frame Snooping

(1) Overview

To allow inspecting specific frames of different protocols for management purposes, a programmable generic frame parsing module is implemented that allows marking a frame for snooping. Snooping allows several functions like e.g. a copy is sent to the management port for inspection while the frame is forwarded normally.

The generic approach allows searching for multiple arbitrary length patterns within a frame to identify (match) a frame for snooping. The following is a non exhaustive list of examples how the generic parsing can be used:

- Mark frames with specific Ethernet type
- Mark frames with specific payload contents
- Mark IP frames with any IP protocol and sub protocol (extension headers) values
- Mark TCP/IP or UDP/IP frames with specific port numbers
- ...

When a frame is marked for snooping, it can be forwarded to the management port exclusively or as a copy or become discarded.

If none of the given rules match, the frame is processed normally. If a frame is coming from the management port itself, the snooping rules are ignored (to avoid the frame be routed back to the management port again). However the management port can use forced forwarding mechanisms to direct any frame to specific output ports if needed.

The function is configured using registers GPARSER[n] and GARITH[n] in the register map and the management port is defined by register MGMT_CONFIG.

(2) Snooping Dataflow Description

The figure below shows the functions involved to mark a frame for snooping. Every frame is inspected, in parallel, with up to 8 individually programmable parsers. These parsers allow comparing a specific pattern from 1 to 16 bits at a certain position within the frame and report a match if the pattern is found.

The match result of each individual parser is then provided to an arithmetic block, which can perform Boolean operations on the result from up to 4 parsers to form a final match decision. Two of such arithmetic modules are available and independently programmable.

When any of the arithmetic blocks reports a match, the final snooping decision for the frame is provided to the switch forwarding.

Figure 4.8 Snooping Function Overview

(3) Generic Parser

Each individual parser can inspect a frame as programmed by register GPARSER[n] (n=0..7). The picture shows the interpretation of the individual settings.

Figure 4.9 Generic Parser Configuration Parameters

The parser acts as follows:

1. It extracts the byte(s) at the given offset of the frame. The offset can either be specified starting after the MAC source address, starting after any optional VLAN tags or starting after an IP header.
2. It then applies a bitmask (AND) to the byte to mask out bits that should be ignored
3. It then compares the result with the given pattern (compare value). If the pattern matches, a match indication is produced.

Instead of the mask&compare function, additional compare options allow to compare against a 16-bit value (e.g. for type comparisons) or alternatively (OR) two different byte values instead.

(4) Parser Limitations

The parser when inspecting frames has limitations. If a limit is reached, the parser is ignored and the frame is forwarded normally as if no parser was active (treated as a “no match”).

The limitations are as follows:

- The bytes that can be inspected can be at maximum at position 67 after the first byte of the frame’s destination address. If the offset together with e.g. VLAN tags falls beyond this limit, the snooping may not have an effect on the frame.
- The bytes within the last 8 bytes of a frame cannot be inspected. That is, if the last byte of the CRC field is 8 bytes or less after the byte to compare.

(5) Arithmetic Function

The match result of four parsers is connected to an arithmetic module combining the results of the parsers to a final match decision.

The arithmetic function implements 4 stages (arith0,1,2,3 in figure below) and Boolean operations at every stage allow to implement complex combinations of the parser results. At each stage, the following can be configured:

- select parser match result (match) or its inverted (i.e. no match)
- select a result of a previous stage (only for the last 3 stages, arith1,2,3)
- define if the selected inputs should be OR'ed or AND'ed and if the result should be inverted

Figure 4.10 Snooping Arithmetic Function

The arithmetic function for the first 4 parsers (0..3) is configured in registers GARITH0..3. The arithmetic function of the next 4 parsers (4..7) is configured in registers GARITH4..7.

The configuration of the last stage (arith3, register GARITH3 or GARITH7) also includes the configuration bits for the snooping function to define if the frame should be forwarded to management or copied or discarded eventually.

(6) Snooping Qualification

The last stage following the arithmetic modules qualifies the results from both arithmetic functions to provide the final decision to the switch how it should process the frame:

1. forward it to the management port only
2. forward it normally and copy it to the management port also
3. discard it

If both arithmetic functions match for the same frame but have different snooping actions configured, the action with the lower number is executed (i.e. forward to management port only before copy before discard).

4.5.3.5 Frame Classification and Priority Resolution

(1) Overview

When a Frame is received on an input port, several information are extracted from the frame as the Ethernet MAC Address, VLAN tag and IP Headers to determine the Frame Type and perform the relevant Classification actions.

In addition, the MAC Address table can provide a priority indication for the destination MAC address if the switch management has programmed the address table accordingly (static entries).

The Frame is classified as a High priority or as a Low priority Frame and is eventually queued in a corresponding priority queue at the output port(s). A higher queue number is given a higher priority.

(2) VLAN Priority Look Up

On each port, an 8-entry programmable priority table is implemented. The registers VLAN_PRIORITY[n] contain the priority mapping for a port [n]. The switch uses 3 bits from the VLAN tag information (3 bits VLAN priority, see **Section 4.5.1.3, VLAN Tag Overview**) to extract the corresponding bits from the Table, which indicates which priority the Frame should be finally classified.

The index in the mapping table is the 3 bits of the first octet of the VLAN Tag Data with bit 5 (Prio(0)) being the LSB and Bit 7 (Prio(2)) being the MSB (see **Section 4.5.1.3, VLAN Tag Overview** for VLAN tag details).

Figure 4.11 VLAN Table Overview

(3) IPv4 and IPv6 Priority Look Up

The switch can classify both IPv4 and IPv6 frames: A 64 Entry Table is implemented per port to classify the IPv4 Frames and a 256 Entry Table is implemented per port to classify IPv6 Frames (The IP Classify Tables) (see **Section 4.4.43, IP_PRIORITY[n] — IP Priority Register [n] (n = 0..4)**).

On the IPv4 Classify table entry, the Frame's 6 bits DiffServ field is provided and the Table returns the 2 bits priority information.

On the IPv6 Classify table entry, the 8 bits Traffic Class field is provided and the Table returns the 2 bits priority information.

CAUTION

The table provides 2 bits for addressing queues 0..3.

Figure 4.12 IP Classify Tables Overview

(4) Priority Resolution

The Priority Resolution function, for a Port [n], is, on each port independently, programmable with the registers PRIORITY_CFG[n] to enable or disable VLAN or IP or MAC address or type based classification.

In addition, the optional Receive Pattern Matcher (see **Section 4.5.14, Receive Pattern Matcher**) can be used to override the classification.

The priority resolution follows the following rule set (and precedence) depending on which classifications are enabled (PRIORITY_CFG[n]) and which fields are found within the frame:

- If the Pattern Match classification is enabled (PATTERN_CTRL.SET_PRIO) and a match occurs, the defined priority from the PATTERN_CTRL register is used, ignoring the settings in PRIORITY_CFG[n].
- Otherwise, if IP classification is enabled and IP header found → map priority according IP_PRIORITY table
- Otherwise, if VLAN classification is enabled and VLAN tag found → map priority according VLAN_PRIORITY table
- Otherwise, if MAC classification is enabled and MAC address found → take priority from address table
- Otherwise, if TYPE classification is enabled and the frame's type (following any VLAN tags) matches an entry of the PRIORITY_TYPE defined values, use the priority from that matching PRIORITY_TYPE
- Otherwise, if it is a BPDU frame transmitted by the management port, and the priority setting in the register MGMT_CONFIG is non zero, this priority is used.
- Otherwise, use default priority as specified in PRIORITY_CFG.

(5) Bridge Control Protocol Identification

To allow for implementation of bridge control protocols like the Spanning tree protocol, an input module is implemented that marks all management frames (Bridge Protocol Data Units, BPDU) when they enter the switch. The mark then can be used by the Input Port Blocking function (see **Section 4.5.3.9(6)(b), Input Port Blocking**) to drop the frame after the address learning.

In addition, the function can be configured to pass all frames or to pass only management frames (e.g. covering spanning tree port states "blocking", "listening", "learning") and discard all other frames.

4.5.3.6 Input Port Selection

The port selection constantly checks (polling) all input ports for available data and if any data is available, a port is selected and frame data is read from the input. All inputs are served simultaneously using a multiplexed access for storing the frame data into the shared memory. No receive FIFOs exist allowing for very low forwarding delays as no copying from a receive FIFO into an output queue is needed.

4.5.3.7 Layer 2 Look Up Engine

(1) Overview

The switch is using the lookup memory to perform address lookups and decide to which port(s) a frame should be forwarded to. By using a port mask, instead of a number, it is possible to forward a frame to one or many destination ports (e.g. multicast). For example, software can add (static) multicast addresses and define that traffic is forwarded only to the ports that have subscribed to the multicast domain.

In addition to the destination lookup, the source address of each incoming frame is searched, and if found, the corresponding aging bit of an entry becomes set. The aging bit is used by the aging process to determine if an entry should be removed from the memory. The aging process will periodically clear the aging bit of each entry and verify if the bit is set again after some time. If the bit is not set, the entry is considered to be outdated and can be removed from the lookup memory.

(2) Hash based Lookup

A hash code is calculated using the frame destination MAC address. It is used as an entry (address) to a table, which contains, for each hash value MAC addresses with destination port mask and validity information.

As one hash code value can represent more than one MAC address, the memory implements for each hash value, up to eight MAC address entries, which are searched linearly.

Figure 4.13 Port Look-Up Overview

(3) Hash Code

For MAC address table 8192 entries, a 10-bit hash value is calculated from the 48-bit MAC address. The hash code is using a CRC-10:

$x^{10} + x^9 + x^5 + x^4 + x + 1$ (233h)

(4) Address Memory

The address memory is divided into blocks. Each block contains eight records, which contain 58-bit of information each. Each record contains the 48-bit MAC address (first byte in bits 7:0) and provides the necessary forwarding information as well as priority or aging information.

Two types of records are defined:

- Dynamic Record:
  The dynamic entry provides the MAC address together with destination port number and aging information. These entries are created by the learning function based on received frames. Dynamic entries are deleted by the aging function if not updated.
- Static Multiport/Priority Record:
  Switch Management can also write static entries in the table, which can include priority information as well as multiple destination ports for forwarding. The MAC address can be unicast or multicast. These records can be used to e.g. specify the ports to participate in a specific multicast domain or to assign a MAC address based priority to a frame. The aging and learning functions ignore static records.

Figure 4.14 Address Memory Record Types

The record's bit 49 decides which type of record is found in the table:

- If 0, a dynamic entry is available and bit 50 holds the aging bit. bits 53..57 indicate at which port the address has been received.
- If 1, a static entry is available with a 3-bit priority field followed by the port bit mask (bit 52 is ignored because the priority is 4 levels).

The record bit 53 represents Port 0, bit 54 Port 1 etc. The frame will be forwarded to all ports that have a 1 in the port bitmask. The source port will be removed dynamically from the bitmask during forwarding (a frame is never forwarded to the port where it came from).

4.5.3.8 Layer 2 Look Up Engine Operational Description

(1) General

The host CPU can be used to configure the lookup options. The switch performs self initialization of the address table after reset for normal operation. Lookup and Learning is performed fully autonomously by the hardware without processor intervention.

It should be noted, that all address table memory transactions from the processor consume time that is lost for the hardware to perform lookups required for frame forwarding. Hence it influences the overall switch performance and therefore should be reduced to a minimum during operation.

(2) Memory Write

Writing to memory is performed in 2 steps.

1. The register LK_DATA_LO/HI are set with the data to write
2. The register LK_ADDR_CTRL is written to start the transaction.

When the transaction is started (by writing into LK_ADDR_CTRL), the interface controller needs some time until the transaction is completed. Software can either poll the busy bit within the LK_ADDR_CTRL register, or set the WAIT_COMPLETE bit when writing to the register. Using the WAIT_COMPLETE bit ensures that the CPU cannot access the registers until the transaction has been completed (the processor bus is held in wait, freezing processor activity).

(3) Memory Read

Reading from the memory is performed in 2 steps similar to writing. However, first the transaction is executed and the data will be delivered to the LK_DATA_LO/HI registers after completion.

1. The register LK_ADDR_CTRL is written to start a READ transaction.
2. Upon completion of the transaction, data is available in LK_DATA_LO/HI.

(4) Power On Initialization

After power on reset, the switch performs self initialization of the lookup table functions and enables automatic learning. The host CPU may perform additional configuration or e.g. add static table entries as needed. The learning functionality and other options are configured with the lookup control register LK_CTRL.

(5) MAC Destination Address Lookup

The 48-Bit destination MAC address of each frame is extracted. If the frame is a Unicast or Multicast frame (See **Section 4.5.1.2** for definition) the look up function can provide three results with three different associated actions performed by the switch hardware:

1. The address is found in the table:
   The switch forwards the frame to the port(s) provided by the port mask of the entry.
2. The address is in the table but is associated with the port on which it was received:
   The switch discards the frame and does not forward it to any port.
3. The address is not found in the table:
   The Switch engine sends the received frame depending on the type of destination address to all ports (flooding) listed in:

- UCAST_DEFAULT_MASK for unicast frames
- MCAST_DEFAULT_MASK for multicast frames

Broadcast frames are always sent to all ports listed in the BCAST_DEFAULT_MASK.

Note that setting any of the default masks to all zero would lead to discard of the frame.

In all cases, the port, where the frame was received from is automatically removed from the final forwarding port mask.

CAUTION

Flooding and additional frame filtering can be further controlled with the mechanism described in **Section 4.5.3.9(3)**.

(6) Automatic Learning / Migration

The hardware allows performing automatic learning of source addresses of incoming frames. When a frame is received from any port, its source address is searched in the lookup memory in addition to the destination address lookup described before.

If learning is enabled (see **Section 4.4.61, LK_CTRL — Learning/Lookup Function Global Configuration Register**), the following tasks are performed by the hardware:

- Checks the INPUT_LEARN_BLOCK register of the switch. If learning is disabled for a port, the source address of incoming frames is ignored.
- If the frame is marked as a management frame (BPDU, see **Section 4.5.3.9(6), Bridge Protocol Frame Resolution**), the source address is never learned (independent of INPUT_LEARN_BLOCK).
- If the source lookup was successful, the port mask is verified to match the incoming port number where the frame was received. If a mismatch occurs, the forwarding port mask is updated accordingly (migration), but only if the entry is a dynamic entry. Static entries are not changed.
  - In case of a redundant port, the port mask is updated adding the redundant port to the existing port mask (OR'ing). This allows automatic learning for single and double attached nodes (see **Section 4.5.12, Parallel Redundancy Protocol (PRP)**).
- If the source lookup failed, the new address is added to the lookup memory (if allowed by configuration in LK_CTRL). If there is no more space in the table, the new entry replaces an arbitrary entry with an identical matching hash value (i.e. one of the 8 entries for the hash is deleted).

- The aging bit of the entry is set. The aging function will continuously scan the table and clear the bit to detect outdated entries.
- An interrupt is generated whenever a source address is not found in the table (this is independent from learning having entered an address or not). See **Section 4.4.62, LK_STATUS — Status Bits and Table Overflow Counter**.

If automatic learning is disabled, no inspection of the frame source address is performed. It is then the responsibility of the software to place the addresses in the lookup memory (static configuration). The learning interrupt will still indicate when new addresses arrive.

Note that management frames (BPDU, see **Section 4.5.3.9(6), Bridge Protocol Frame Resolution**) are never learned by the hardware and it is the responsibility of the software to add a source address of such frames if necessary.

(7) Aging

Aging refers to deleting old entries within the table. It proceeds as follows:

1. For each entry, an aging bit is implemented. The bit is set whenever the source address is found within a received frame.
2. When the aging process (timer) inspects an entry, it resets the aging bit to 0. If it finds an entry with the aging bit already being 0, the entry is deleted.
3. Static entries are not affected by the aging process and kept always.

This process runs continuously in the background processing one entry at a time. The aging period is programmable (see **Section 4.4.67, LK_AGETIME — Period of the Aging Timer**), giving a range from seconds to several minutes. Aging can also be disabled completely under software control.

4.5.3.9 Frame Forwarding Tasks

(1) Overview

When an input port has been selected, the frame is forwarded to its corresponding output port(s) as necessary. Output port resolution and switching is based on the information from the two stage MAC destination address lookup followed by additional resolution functions to allow frame duplication and flooding control, which are described in the following sections.

Figure 4.15 Frame Forwarding Tasks Overview

(2) VLAN Domain Verification

When a frame is received at a port, it can be verified to be within the correct VLAN domain. The register VLAN_VERIFY is used to define if input verification should occur. Output verification is always implicitly done by the VLAN resolution function, which will forward a frame only to ports that are a member of the VLAN as defined by the VLAN table (see **Section 4.5.3.9(3)(b), VLAN Domain Resolution / VLAN Table**). The following rules apply if the verify bit within VLAN_VERIFY for the port is enabled:

- If the frame's VLAN id is found within the table and the input port is member of the VLAN domain, the frame will be forwarded normally.
- If the frame's VLAN id is found within the table but the input port is not a member of the VLAN domain, the frame will be marked invalid and will be discarded eventually.
- If the frame's VLAN id is not found in the VLAN table, or the frame has no VLAN tag, the frame will be forwarded normally, or, if the discard bit for the port is set (also in register VLAN_VERIFY), it will be discarded.

CAUTION

When the VLAN_VERIFY does not enforce input verification, the output verification (VLAN domain resolution) is still active always. To avoid unexpected behavior if VLAN domains are not used, the complete VLAN_RES_TABLE should be programmed to have an all 1 port mask in all entries.

(3) Broadcast / Multicast / VLAN Domain Resolution

(a) Overview

To ensure that traffic within VLAN channels are always routed to the correct ports, for example to avoid the duplication of critical information through a network, the switch implements a resolution mechanism that, for any frame that is switched to multiple ports, checks the VLAN ID provided with the current frame.

The VLAN resolution mechanism searches the VLAN Resolution Table (see **Section 4.4.51, VLAN_RES_TABLE[n] — 32 VLAN Domain Entries (n = 0..31)**), which stores up to 32 unique VLAN IDs, each associated to a port bit mask. It is used to limit forwarding of frames only to ports that are member of a VLAN domain.

(b) VLAN Domain Resolution / VLAN Table

The VLAN resolution table (registers VLAN_RES_TABLE) provides a unique VLAN ID / port bit mask association for up to 32 VLANs. A default entry (register BCAST_DEFAULT_MASK) provides an additional port bit mask. The port bit mask implements one bit for each port.

Each Port Bit indicates, if set to "1", that it is member of the VLAN and frames with the corresponding VLAN ID can be switched to the port. If the Port Bit is set to "0", a frame with the corresponding VLAN ID will not be switched to that port. If no VLAN ID matches, the default mask is applied.

Figure 4.16 VLAN Resolution Table Overview

CAUTION

The port mask is found in bits [4:0] and the VLAN info is found at bits [16:5].

Note that the VLAN table cannot be deactivated. To avoid unexpected behavior if VLAN domains are not used, all entries of the table (i.e. all VLAN_RES_TABLE registers) should be programmed to have an all 1 port mask in all entries (reset default).

(c) VLAN Switching / Resolution Mechanism

The VLAN table is used for both, VLAN domain verification (see **Section 4.5.3.9(2), VLAN Domain Verification**) as well as VLAN resolution. Once the frame has passed any VLAN domain verification (i.e. will not be discarded by the verification function already), the forwarding resolution applies.

- If the destination MAC address (Unicast or Multicast) is found in the MAC address table:
  - If the frame carries a VLAN tag that is found in the VLAN table, the frame can be forwarded only to the ports within the VLAN domain and will be discarded if the destination port is not member of the VLAN domain.
  - If the frame carries a VLAN tag that is not found in the VLAN table, or does not contain a VLAN tag, it is forwarded as indicated by the lookup table (note that VLAN domain verification can be configured to discard the frame in this case if enabled).
- If the destination MAC address (Unicast or Multicast) is not found in the MAC address table, or if the destination address is the Broadcast address, the frame is forwarded according to the following rules:
  - The destination port mask is loaded from the respective register U/M/BCAST_DEFAULT_MASK depending on unicast, multicast or broadcast. Then the following filtering on this mask applies.
    - If the frame carries a VLAN tag, the VLAN resolution table is searched for a matching VLAN ID and the frame is sent only to ports that are associated with the VLAN ID.

- If the frame carries a VLAN tag and the VLAN ID does not match any entry in the VLAN Resolution Table, or the frame does not carry a VLAN tag, the frame is forwarded to all ports that are enabled by the default mask.
- If it cannot be associated with any VLAN group and if the default group has been set to all zero, the frame is discarded.

To disable the VLAN resolution, set all VLAN IDs to 0x000 and all Port Mask bits to “1”. If the VLAN resolution is disabled, normal port flooding is implemented as described in **Section 4.5.3.8(5), MAC Destination Address Lookup**. The default entry can still be used to restrict flooding to only dedicated ports, if not programmed to all “1”.

(4) Port Mirroring

The optional function allows duplicating traffic to a dedicated mirror port. Any one of the ports can be assigned to act as a mirror port (register MIRROR_CONTROL).

The mirror port then is always added to the list of output ports and therefore receives a copy of the frame, if any of the following rules matches with the currently processed frame:

- Ingress Port Number Match
  When a frame is received on port[n] and the corresponding bit in the register MIRROR_ING_MAP is set to 1, the frame is mirrored.
- Egress Port Number Match
  When a frame is forwarded to port[n] and the corresponding bit in the register MIRROR_EG_MAP is set to 1, the frame will be mirrored.
- MAC Ingress SA Match
  When the Ingress Port Number match succeeded (see above) and the MAC source address matches the MIRROR_ISRC, the frame will be mirrored.
- MAC Ingress DA Match
  When the Ingress Port Number match succeeded (see above) and the MAC destination address matches the MIRROR_IDST, the frame will be mirrored.
- MAC Egress SA Match
  When the Egress Port Number match succeeded (see above) and the MAC source address matches the MIRROR_ESRC, the frame will be mirrored.
- MAC Egress DA Match
  When the Egress Port Number match succeeded (see above) and the MAC destination address matches the MIRROR_EDST, the frame will be mirrored.

In addition, a counter is implemented (register MIRROR_CNT) that allows specifying that only every Nth frame that matches any of the above criteria is mirrored. If the counter is set to 1 or 0, every frame is mirrored that matches any of the above criteria.

(5) Congestion Resolution

(a) Overview

The congestion resolution function is used whenever an output port is not available and data needs to be sent to that port. An output port is defined to be available if the port is enabled (bit n in register PORT_ENA.TXENA is “1”, n = 0..4) and its output queue is not full. If a port is deasserted (bit n in register PORT_ENA.TXENA is “0”), the port is not available and Frames cannot be switched to that port.

The congestion resolution function determines whether the frame should be processed further or discarded according to the following rules.

(b) Unique Destination (one input to one output)

If the output port is enabled and can accept, a frame the frame will be forwarded normally.

In any other case, the frame will be discarded.

(c) Multiple Destinations (Flooding)

After broadcast / flooding resolution, a frame needs to be switched to multiple output ports.

- Output disabled:
  All disabled ports are removed from the list of outputs.
- Output congestion:
  If any of the outputs cannot accept a frame (as indicated by the output queue management for the port, implementation specific), it is removed from the list of outputs also.

If, after the removal no output port is left in the list of outputs, the frame is read from the input and discarded. This event is counted in the port specific ODISC statistics counter.

(6) Bridge Protocol Frame Resolution

To implement bridge control protocols like the Spanning Tree protocol, the following control functions are performed by the Bridge Protocol Frame Resolution function.

(a) Management Frame (BPDU) Identification

Bridge Protocol Frames are identified by its destination address being any of the following:

- 01-80-c2-00-00-00 to 01-80-c2-00-00-0F (*Spanning Tree, IEEE 802.1D, Table 7-9*)
- 01-80-c2-00-00-10 (*Bridge Management Address, IEEE 802.1D, Table 7-10*)
- 01-80-c2-00-00-20 to 01-80-c2-00-00-2F (*Generic Attribute Registration Protocol, IEEE 802.1D, Table 12-1*)

In addition, the following features are available to define a frame as Bridge Protocol frame which will then experience the same treatment as any other BPDU within the switch.

- Custom Address:
  To support other protocols and standards, the switch allows defining one additional address (see **Section 4.4.47, MGMT_ADDR0_lo — Lower MAC Address for Bridge Protocol Frame** and **Section 4.4.48, MGMT_ADDR0_hi — Higher MAC Address for Bridge Protocol Frame**) which is treated identically to the above, marking a frame as a Management (BPDU) frame.

To support IEEE 802.1X authentication, the destination address 01-80-c2-00-00-00 is used by EAPOL frames and can be treated differently from BPDU frames (see **Section 4.5.8, Port Based Access Control (802.1X)**).

(b) Input Port Blocking

The input port blocking function is used to avoid forwarding of frames after address learning. The firmware can program the register INPUT_LEARN_BLOCK and if a frame is received on a port [n] which should be blocked (blocking bit n = 1) and the frame is not a bridge protocol frame, the frame will be marked for discard and will not be forwarded to any output port.

(c) Input Port Learning Disable

To reduce processing load from the firmware, a port can be configured for exclusion from learning (see **Section 4.4.8, INPUT_LEARN_BLOCK — Input Learning Block Register**). When learning is disabled on a port, no source address extraction happens for incoming frames.

Incoming BPDU frame source addresses are never extracted and not forwarded to the learning process independent from this setting. The application (management port) will need to add source addresses from received BPDUs to the lookup table e.g. as static entries as needed.

(d) Management Frame Forwarding

If a frame is coming from any port except the management port and is marked as a management frame (BPDU frame) and the management port is enabled (see **Section 4.4.9, MGMT_CONFIG — Management Configuration Register**), the frame is forwarded to the management port only, ignoring any destination address lookups.

If the management port itself receives management frames (i.e. host CPU sends management frames to switch), they are forwarded according to the port mask defined in the configuration register MGMT_CONFIG. A handshaking mechanism is implemented that can be used by the protocol software to configure the destination port mask on a frame by frame basis for management frames.

In addition, the port mask given in MGMT_CONFIG can include ports that are disabled for normal traffic (configured in register PORT_ENA.TXENA) allowing transmitting management frames on ports that are disabled for normal traffic.

When forced forwarding is used, management frames will be forwarded to the ports given in the forced forwarding mask and the port mask given in MGMT_CONFIG is ignored.

NOTE

VLAN domain verification/discard (see **Section 4.4.5, VLAN_VERIFY — Verify VLAN Domain**) should be switched off for the management port to avoid that the switch discards management frames.

(7) Forced Forwarding

When the forced forwarding is used, it overrides any forwarding decision and a frame will be forwarded to all ports given in the forced forwarding mask.

This includes forwarding to disabled ports (i.e. overriding PORT_ENA) when the force filtering is disabled (0). Only the port where the frame was received from will be removed from the port mask automatically (i.e. it is not possible to transmit a frame to where it was received from).

Mirroring and Snooping functions may continue to inspect forced forwarded frames.

(8) Cut Through Forwarding

The switch normally operates in so-called store and forward mode. All frames are received and stored completely before they are forwarded to any output port. This allows error detection and discard of erroneous frames within the switch as well as communication between links of different speeds.

Alternatively, the switch can operate in cut through mode, where a frame will be forwarded and transmitted at an output port before it is completely received. This allows for reduced latency and is independent of the frame's size. However as the frame is forwarded before error checks can be done, erroneous frames will become transmitted (however the switch ensures to mark them erroneous and sends it with an invalid CRC).

Cut Through operation is configurable on a per port basis with register MODE_CONFIG. It allows to define a group of ports for cut through and have other ports using normal store&forward.

When cut through forwarding is enabled, a frame is forwarded using cut through only if the following conditions are all valid. Otherwise the frame is forwarded using store&forward:

- Destination lookup resolves only a single destination port (i.e. no broadcast or flooding that would require forwarding to multiple destination ports).
- The source and destination ports have both their cut through mode bit set.
- It is not a broadcast, not a BPDU, not a IEEE 1588 PDELAY_REQ frame (auto-response).
- The source and destination port operate at the same link speed.

It means only 1:1 forwarded frames are able to use cut through. If cut through is not possible for a frame, it is forwarded using store&forward as normal.

The cut through forwarding delay depends on link speed and the switch system clock frequency. For a switch system clock of up to 200 MHz, the following forwarding delays are possible.

Table 4.228 Link Speed vs Forwarding Delay

Link Speed	Forwarding Delay (SFD to SFD, without PHY latencies)
10 Mbps	Up to 35 µs
100 Mbps	3.2 .. 3.8 µs (no pattern matcher active) 3.5 .. 3.9 µs (pattern matcher active)
1 Gbps*1	550 .. 800 ns (pattern matcher active)

Note 1. Using cut-through with Gigabit ports can cause TX FIFO underflows corrupting outgoing frames. To avoid this, gigabit ports should also have at least one RX Pattern Matcher enabled as this will delay the cut-through decision slightly (by up to 4 bytes) preventing the underflow. The pattern is not relevant and does not need to match anything (e.g. an all 0 pattern with all 1 mask never matches), but will still delay the forwarding decision by the required extra time.

(9) Switching

After the output port(s) have been determined, the switch control enables the corresponding path through the Switch Matrix and the frame is forwarded to the output queue(s).

In a similar fashion, if a frame should be switched to multiple ports (e.g. Broadcast), the switch control enables the corresponding paths though the Switch Matrix and the frame is forwarded to all the destination output ports. Duplication is eventually done by the output ports reading the same frame from memory multiple times.

4.5.3.10 Broadcast Storm Protection

To avoid network and memory congestion caused by a broadcast storm, the switch engine can be programmed to limit the number of broadcast frames over a certain time period.

A counter is implemented on each receive port, which increments whenever a broadcast frame is received. The counters of all ports are cleared each time the timeout expires.

When the counter of a port reached the limit, further broadcast frames received at this port are discarded.

The register BCAST_STORM_LIMIT is used to set the periodic timeout and the counter limit for all ports.

4.5.3.11 Multicast Storm Protection

To avoid network and memory congestion caused by a multicast storm, the switch engine can be programmed to limit the number of multicast frames over a certain time period.

The behavior is identical to the broadcast storm protection, however implements its own per port counters.

The register MCAST_STORM_LIMIT is used to set the counter limit for all ports. The periodic timeout from BCAST_STORM_LIMIT is used to reset the counters.

4.5.3.12 Output Frame Queuing

The following is a conceptual Output Queue management function supporting multiple Queues.

The implementation offers 4 queues numbered 0..3 where 0 is lowest and 3 is highest priority.

The output queues (FIFOs) are implemented separately for each output port. After input processing has completed, the frame is stored in a queue within the corresponding output port, based on the priority resolution result for the frame (Classification, see **Section 4.5.3.5(4), Priority Resolution**).

Figure 4.17 Output Queuing

The Queue Arbitration selects the appropriate Queue and indicates data availability to the output interface, which eventually will transmit the frame. A higher queue number takes precedence over a lower queue number.

The Queue Arbitration implements a strict priority selection. Weighted fair queuing is also available (see **Section 4.4.37, IMC_CONFIG — Input Memory Controller Configuration Register**).

In addition, the optional TDMA scheduler influences arbitration by allowing read from a queue only at specific times (see **Section 4.5.15, TDMA Operation**).

The MAC connected to the port's data interface will eventually de-queue the frame at its local transmit port speed.

4.5.3.13 Output Port Congestion Management

Depending on the output queue management method, a congestion indication is generated to the switch (internally) when a queue becomes full. When a frame needs to be forwarded to a congested queue, the frame will be discarded and the according error counter (Switch statistics ODISC) will be incremented.

(1) Tail Drop

By default, frame discard occurs only when a queue becomes full, or if the shared switch memory is exhausted. Frames will be dropped as long as the congestion situation persists.

The implementation allows to store up to 58 frames per queue per port. This is independent of the frame size.

(2) Random Early Detection

As an alternative to Tail Drop the Random Early Detection (RED) active queue management following RFC 2309 is available (**Section 4.6.2**, (1)). It can be enabled per queue but only globally for all ports (see registers QMGR_RED_MIN4*[1]/MAX4*[2]/CONFIG*[3]).

The RED algorithm averages the fill level of a queue over time. Depending on the current average a drop probability is calculated and frames are dropped before the queue becomes full. When filled completely, tail drop occurs. The early dropping of frames may help higher protocols (mainly TCP) to better adapt to the available link bandwidth. The algorithm is configured using two thresholds defining a window where frames will be dropped before the queue's capacity is reached. No drop occurs below the minimum threshold and all frames are dropped when the maximum is reached (resulting in tail drop behavior). Between minimum and maximum the probability for dropping a frame increases. That is, the more the queue is filled the more frames will be dropped which, ideally, avoids reaching the maximum which causes continuous frame drops.

When the current queue fill level drops below the calculated average, the average is immediately set to the current level of the queue. This prevents a delayed reaction where the queue is already emptied again while the average would still indicate a higher fill level causing unnecessary dropping of frames.

Note 1. See **Section 4.4.30, QMGR_RED_MIN4 — RED Minimum Threshold Register**.
Note 2. See **Section 4.4.31, QMGR_RED_MAX4 — RED Maximum Threshold Register**.
Note 3. See **Section 4.4.32, QMGR_RED_CONFIG — RED Configuration Register**.

The following figure illustrates the behavior.

Figure 4.18 Random Early Detection Algorithm Implementation

The algorithm avoids dropping of frames when the queue is draining which means the output rate is higher than the input rate and hence dropping of frames should not occur. This allows to absorb a short peak in the queue size making use of the available storage capacity, ideally without (unnecessary) dropping activity.

4.5.4 Timestamping Functions (TSM)

4.5.4.1 Overview

The timestamping module (TSM) implements the following functions:

- collects a timestamp for received frames on all line ports.
- collects a timestamp for transmitted frames on all line ports with interrupt generation.
- processes IEEE 1588 Layer 2 event frames updating on the fly (1 step) the correction field with the transient time.

4.5.4.2 IEEE 1588 Message Header Structure

CAUTION

The following is provided as a brief reference only. Please refer to the standard documents for details and usage scenarios.

An IEEE 1588 message can be transported with various encapsulation protocols for example within an Ethernet frame (L2) payload or an UDP/IP payload. The following figure shows the message header, which is common to all IEEE 1588 version 2 messages. The MAC allows updating the correction field during transmission when using Layer 2 frames which are identified by Ethernet type field being 0x88f7. Other encapsulations are not supported. Correction field updates only occur in event messages (i.e. messageId < 4).

All fields follow the network byte order which is the first byte transmitted is the most significant byte (leftmost) of a multi byte field. The following tables provide a short overview. For details of any of the fields, please refer to the IEEE 1588 specification document (**Section 4.6.2**, (2)).

Table 4.229 IEEE 1588v2 Message Header (PTPv2)

Bits								Octets	Offset
7	6	5	4	3	2	1	0		
transportSpecific				messageId				1	0
reserved				versionPTP = 0x2				1	1
messageLength								2	2
domainNumber								1	4
reserved								1	5
flags								2	6
correctionField								8	8
reserved								4	16
sourcePortIdentity								10	20
sequenceId								2	30
control								1	32
logMeanMessageInterval								1	33
further message specific fields may follow								n	34

The type of message is encoded in the field messageId as follows (informal):

Table 4.230 PTPv2 Message Type Identification

messageId	Message Name	Message
00h	SYNC	event message
01h	DELAY_REQ	event message
02h	PDELAY_REQ	event message
03h	PDELAY_RESP	event message
04h – 07h		reserved
08h	FOLLOW_UP	general message
09h	DELAY_RESP	general message
0Ah	PDELAY_RESP_FOLLOW_UP	general message
0Bh	ANNOUNCE	general message
0Ch	SIGNALING	general message
0Dh	MANAGEMENT	general message

4.5.4.3 Timestamp Processing

(1) Receive Timestamp Processing

When a port receives a frame, the timestamp, based on the current time from the timer, is captured when the start of frame delimiter (SFD) is detected at the PHY interface. The timestamp is forwarded together with the frame throughout the switching engine and can be accessed by the management port to implement e.g. the Precision Time Protocol (PTP).

The receive timestamp is encapsulated in the proprietary tag information for frames forwarded to the management port (see **Section 4.5.5, Management Port Specific Frame Tagging**).

(2) Transmit Timestamp Processing

When a frame is transmitted to the PHY on a port, the timestamp is also captured. The outgoing timestamp can be stored in the port specific timestamp register (PORT[n]_TIME) for each port. The management port, through its special control information, has the capability to identify frames for outgoing timestamp capture, as timestamps must only be captured for dedicated event frames, not for all frames.

4.5.4.4 Transparent Clock Support

(1) General

The hardware implements the necessary functions to implement so-called transparent clocks (TC) in both variants,

a) the end to end variant and

b) the peer to peer variant.

The end to end TC is the default behavior when the register PEERDELAY is set to 0.

When a peer to peer transparent clock is implemented, the PTP software has to determine the peer delay at the port where the master is connected to (where it receives SYNC messages). This value then is programmed into the PEERDELAY register(s). The setting needs to be updated when a new peer measurement has been done or whenever the master port changes.

(2) Implementation of Correction Field Update

The implementation is capable of updating the correction field within Layer 2 PTP frames only (i.e. frames with type 0x88f7). PTP messages within UDP/IP frames are not considered.

The update module only processes event messages, which it automatically detects from the message type field found within the PTP header (type < 4). This means that it does not process follow up frames (which are not event frames), hence any correction field updates will be found in the corresponding SYNC frame. This allows supporting one step as well as two step master and slave nodes connected to the switch.

For an end to end transparent clock implementation, the correction field of SYNC and DELAY_REQ messages is updated with the transient time (output time - input time).

For a peer to peer transparent clock implementation, the correction field update includes adding the value set in the PEERDELAY register (of the port where the frame was received) to the transient time. The hardware always implements the addition of this value, hence it only depends on configuring the PEERDELAY to define whether an end to end (PEERDELAY = 0) or peer to peer (PEERDELAY > 0) behavior is selected. It is the responsibility of the PTP software to exchange the PDELAY_REQ/PDELAY_RESP messages on every port to determine the peer delay values accordingly.

Correction field updates occur only on frames that are exchanged in between external (line) ports. Any frame from and to the management port will not be modified.

The following figure shows the operations when a frame is forwarded in between line ports.

Figure 4.19 One-Step Correction Field Update

CAUTION

No specific configuration is necessary (aside from PEERDELAY register and having a correctly running timer). The function is always active and automatically detects IEEE 1588v2 Layer 2 event frames for processing.

4.5.4.5 Using One Step update at the Management Port

When sending frames through the management port to the switch, the normal correction field updates are not available as there is no notion of a “received” timestamp at this interface.

However to allow master or boundary clock implementations that want to use one step precise timestamping for locally generated event frames (e.g. SYNC messages), the management tag one step update feature can be used (see **Section 4.5.5.4, Receive Processing (CPU to Switch)**, ControlData[4]).

To use this function, an event frame requires special preparation by the application: The correction field must be set to the same value as the originTime field’s nanoseconds value. When preparing a frame for one step update, the application needs to perform the following steps:

- Read out the current time (seconds, nanoseconds) from the hardware timer.
- Prepare the event frame (e.g. SYNC message) and copy the time values retrieved from the timer into the originTime field of the frame. The originTime field contains seconds and nanoseconds fields.
- Set the frame’s correction field nanoseconds value the same as the originTime nanoseconds (seconds are not relevant to the correction field).
- Transfer the frame to the switch using the management tag feature and setting ControlData[4] to 1 (see **Section 4.5.5.4, Receive Processing (CPU to Switch)**).

The MAC transmitter when transmitting the frame will rewrite the correction field with the difference of the transmit timestamp and the value in the correction field. Note that this function does not add to the correction field, it reads the value, subtracts it from the transmit time and writes the result back into the correction field (after performing a 1 second modulo). Any PEERDELAY register settings are ignored. The following figure shows the principle of operations.

Figure 4.20 Management Port One-Step Transmit Feature

A receiver will always add the correction field value to the originTime field’s value and hence will get the precise transmit time of the frame as:

T = To + Cr which is To + Ttx − To = Ttx.

CAUTION

As the update function does not modify the seconds value within the frame, the application must ensure the time delay from reading the hardware timer value and the actual frame transmission at the switch output port is less than one second.

4.5.4.6 Automatic Response to Peer Delay Request Messages

(1) General

Each line port MAC can be configured to automatically respond to incoming peer delay request (PDELAY_REQ) messages. These are used in peer to peer clock implementations to measure the link delay between neighbor nodes. Both nodes of a link periodically transmit the PDELAY_REQ message to each other to allow both nodes to determine the link delay.

A node when receiving a PDELAY_REQ message responds with a PDELAY_RESP message. This enables the requesting device to calculate the path delay.

The switch management application will periodically issue PDELAY_REQ messages to determine the link delay on each of its line ports to set each port's PEERDELAY registers. The PEERDELAY register is used by the transparent clock when forwarding SYNC frames through the switch (see **Section 4.5.4.4, Transparent Clock Support**).

In addition, the switch will receive PDELAY_REQ from all its connected neighbor nodes and needs to respond with PDELAY_RESP messages, individually per port. The management application can be used to create such response messages, but as incoming requests have no relevance to the local system this only represents additional processing overhead.

(2) Response Generation

To reduce overhead, each line port MAC can be instructed to automatically answer with a PDELAY_RESP message to incoming PDELAY_REQ messages. The incoming request is converted to a response and routed through the switch, however will be transmitted only at the port where it was received (i.e. emulating a MAC local response). This function can be enabled in the PTPAutoResponse_P[n] register within a port's MAC register set.

When autoresponse is enabled, the following functions are performed:

- A PDELAY_REQ message is identified by an IEEE 1588v2 Layer2 frame (type 88f7h) with messageId = 2 (1st octet in PTP header, see **Section 4.5.4.2, IEEE 1588 Message Header Structure**).

The next steps occur only when a PDELAY_REQ has been identified:

- If the frame's destination address is unicast, the incoming frame's source address will be copied into the frame's destination address. Otherwise the destination address is not modified (i.e. if the neighbor sends the request with multicast the response will return to the same multicast)
- The frame's source address is set to the port's MAC address (register MAC_ADDR_[01]_P[n])
- Address lookup and learning is suppressed for the frame. The frame will be transferred to the switch but is marked for exclusive forwarding to the same port (this is an exception of the normal forwarding rules which do not forward to the same port). Hence normal statistics, etc. will occur as with any other frame.
- When the frame is forwarded, it is converted into a one-step PDELAY_RESP by the MAC transmitter according to **Section 4.6.2, References**, (2) *11.4.3 b* as follows:
  - The messageId is changed to 3 (PDELAY_RESP).

- The twoStepFlag bit within the flags field of the header is cleared (bit 1 of first octet, see **Section 4.6.2, References**, (2) 13.3.2.6) to indicate a one-step response.
- Adds the transient time (i.e. not considering the port's PEERDELAY[n] register value) to the correction field.
- The sourcePortIdentity is set according to the values from registers PTPClockIdentity1_P[n], PTPClockIdentity2_P[n], PTPAutoResponse_P[n][31:16] as follows:

Table 4.231 PTP AutoResponse Header SourcePortIdentity Mapping

Register[Octet]	Octet Offset in PTP header, Name	
PTPClockIdentity1_P[n][7:0]	20, portIdentity.ClockIdentity[0]	
PTPClockIdentity1_P[n][15:8]	21, portIdentity.ClockIdentity[1]	
PTPClockIdentity1_P[n][23:16]	22, portIdentity.ClockIdentity[2]	
PTPClockIdentity1_P[n][31:24]	23, portIdentity.ClockIdentity[3]	
PTPClockIdentity2_P[n][7:0]	24, portIdentity.ClockIdentity[4]	
PTPClockIdentity2_P[n][15:8]	25, portIdentity.ClockIdentity[5]	
PTPClockIdentity2_P[n][23:16]	26, portIdentity.ClockIdentity[6]	
PTPClockIdentity2_P[n][31:24]	27, portIdentity.ClockIdentity[7]	
PTPAutoResponse_P[n][31:24]	28, portIdentity.PortNumber[0]	(msb)
PTPAutoResponse_P[n][23:16]	29, portIdentity.PortNumber[1]	(lsb)

- The requestReceiptTimestamp field (10 octets at offset 34) is set to 0.
- The frame's original sourcePortIdentity field is copied into the requestingPortIdentity field (10 octets at offset 44).
- All other bits/fields of the request are left unchanged and will hence be copied unmodified into the response message.
- The CRC of the message is replaced accordingly.

(3) Usage Notes

- Auto Response generation is available only for Layer 2 frames (i.e. not with UDP/IP).
- Auto Response forces the MAC RX to strip the CRC. The TX will append a new CRC. Ignores the COMMAND_CONFIG_P[n].CRC_FWD configuration setting.
- When using the recommended PTP multicast destination address (01-80-c2-00-00-0e), the register MGMT_CONFIG must have management enabled, or at least the BPDU discard bit cleared. Otherwise the frame is discarded as this is a BPDU address.
- The normal priority resolution functions apply to the frame when forwarding. The PRIORITY_TYPE registers can be used to set the priority of Layer 2 PTP messages.
- The destination MAC address change is performed at the MAC RX. Hence if mirroring is active, it would show the frame with the already modified destination address.
- Although not required by PTP, a single VLAN tag is supported for PTP messages.
- If the port has VLAN input/output manipulation functions active, those settings are ignored for the response generation.
- Auto Response is supported on PRP group ports (as a response is link local, no trailer processing or duplication is performed).

4.5.5 Management Port Specific Frame Tagging

To support the CPU through a normal networking interface (GMII), some special side band information required for implementing management protocols require the addition of such information into frames exchanged between the CPU and the switch. This function is available only on the dedicated management port when a MAC is implemented there.

Management frames are frames that are identified by their destination address or type as so-called Bridge Protocol Data Units (BPDU) as defined in **Section 4.5.3.9(6), Bridge Protocol Frame Resolution** (destination 01-80-c2-00-00-xx).

Instead of only affecting management frames, optionally, the function can be configured to include the side band information into all frames transferred from the switch to the CPU.

The CPU has the flexibility to insert control information into any frame it transfers to the switch. The hardware inspects all frames received at the management port and will remove the extra data as necessary.

4.5.5.1 Tagged Frame Format

The additional control information is added following the frame source address marked by a programmable (proprietary) ether type (ControlTag). If the tag exists, it is always the very first in the frame (i.e. before any VLAN tags).

The following information is added:

- ControlTag:
  identifier that the control data is present within the frame (default: E001h)
- ControlData:
  2 octets following the ControlTag with control information
- ControlData2:
  4 additional octets used to carry 32-bit frame timestamps or destination port masks depending on transmit or receive.

All data following the ControlData2 are from the original frame. For example, any VLAN tags will be found in this portion of the frame. The control tag, if present, is always the very first tag of a frame.

Table 4.232 Management Port Frame Tagged Frame Format

7 octets	PREAMBLE
1 octet	SFD
6 octets	DESTINATION ADDRESS
6 octets	SOURCE ADDRESS
2 octets	ControlTag (default: E001h)
2 octets	ControlData
4 octets	ControlData2 (timestamp, portmask)
2 octets	type/length
0..1500/9000 octets	PAYLOAD DATA
0..42 octets	PAD
4 octets	FRAME CHECK SEQUENCE

Note that the frame is extended by 8 bytes and the receiving MAC must be enabled to accept such enlarged frames as necessary.

As the tagging function changes the frame contents, the MACs should be configured to strip CRC at the receiving ports. The management port's transmitter will eventually append a new CRC.

4.5.5.2 Byte Order

The first octet of ControlData is the MSByte (bits 15:8) and the 2nd octet of Control Data is the LSByte (7:0) of the 16-bit data field.

The first octet of ControlData2 is the MSByte (bits 31:24) and the 4th octet of ControlData2 is the LSByte (7:0) of the 32-bit data field.

4.5.5.3 Transmit Processing (Switch to CPU)

When the switch forwards a frame to the management port to the host CPU, the following information is added into management (or all) frames.

Table 4.233 Management Frame Transmit Tagging

Field	Bit#	Description
ControlData	b3 to b0	Port number where the frame was received from. Ports are enumerated starting with 0 to 3.
	b15 to b4	Reserved, unused
ControlData2	b31 to b0	Received timestamp of the frame. The 32-bit nanoseconds value when the start frame delimiter (SFD) was detected at the port where the frame was received.

4.5.5.4 Receive Processing (CPU to Switch)

When the CPU sends a frame to the switch, the tagging function checks the frame and removes the optional control tag information from the frame if present. The CPU is free to add the tag to a frame as necessary or leave it out for normal frames that do not need special treatment.

When the tagging function detects the control tag, it removes all 8 octets from the frame before it hands the frame on to the switch.

The control tag information for received frames provides the following information.

Table 4.234 Management Frame Receive Tagging Fields (1/2)

Field	Bit#	Description
ControlData	b0	Forced Forwarding If set, the frame is forwarded to all ports as defined in the destination port mask defined in ControlData2. If 0, the forwarding mask is ignored and the frame is forwarded normally.
	b1	Forced Forwarding, include filtering. If set together with forced forwarding, normal filtering of the destination port mask applies (i.e. disabled ports will be removed from the list). If 0, the frame is forwarded also to disabled ports. **Note)** This applies to BPDU frames only. Normal frames will be filtered always (i.e. can never be transmitted to disabled ports).
	b2	Unused, set 0 always
	b3	Mark frame for transmit timestamping. When set, the frame transmit timestamp will be latched into the corresponding port's transmit timestamp register, when the frame is transmitted. See **Section 4.5.4, Timestamping Functions (TSM)** for details on timestamping functions
	b4	Mark frame for special one-step correction field update (option). When enabled, the correction field of the frame is subtracted from the tx timestamp upon transmission and the result replaces the correction field. The application needs to prepare the frame specifically to use this update method as described in chapter Transparent Clock Support)

Table 4.234 Management Frame Receive Tagging Fields (2/2)

Field	Bit#	Description
ControlData	b5	Suppress PRP trailer append. If 1, suppresses the switch adding a redundancy control trailer (RCT) when sending the frame at a port defined in the PRP group (i.e. bypassing the RedBox function). The bit has no effect if transmission occurs at a normal port. If 0, the switch can append the RCT as needed. Must be 0 if ControlData[6] = 1.
	b6	Force PRP trailer append. If 1, indicates that the transmitter must append a redundancy control trailer (RCT) when sending the frame at a port defined in the PRP group. This enforces trailer append, even if the frame is not duplicated to both redundant networks. Depending on bit 7 either the switch will generate a sequence number, or the application can provide the sequence number that should be used. The bit has no effect if transmission occurs at a normal port. If 0, no trailer append request is given. The bit has an effect only, if ControlData[5] = 0 and is ignored otherwise.
	b7	Use provided sequence number for the PRP trailer. If 1, indicates that the sequence number for the frame to be used when appending the RCT is found in ControlData2[31:16]. If 0, the switch will automatically generate a sequence number when forwarding the frame to the PRP ports. This requires that the source address of the frame is stored in the MAC address table (either a dynamic or static entry). The bit has an effect only, if ControlData[6] = 1 and is ignored otherwise. When using a provided sequence number (i.e. bit[6,7] = 11b), the sequence number/history memory is not updated. Hence if a following frame would be sent without this method, an arbitrary/older sequence number from the memory is added (this may be useful for testing out-of-sequence error scenarios)
	b15 to b8	Unused, set 0 always
ControlData2	b3 to b0	Destination port mask. Relevant only if ControlData forced forwarding bit (bit 0) is set. It then presents the list of ports where the frame should be forwarded to. One bit per port. A bit set to 1 will cause the frame being forwarded to that port. Bit 0 = port 0, Bit 1 = port 1, ... The management port bit is automatically removed from the list during switch forwarding, hence it is not possible to send a frame back to the management port itself.
	b15 to b4	Reserved, unused
	b31 to b16	When ControlData[7] = 1, the 16-bit sequence number for use with the frame when appending the redundancy control trailer (RCT). Reserved otherwise.

NOTE

The tag fields are inserted into the frame in network byte order. That is ControlData[15:8] followed by ControlData[7:0] then ControlData2[31:24] first until ControlData2[7:0] last.

4.5.5.5 Module Configuration

The tagging module is configured by the register MGMT_TAG_CONFIG.

The module can either insert the control information into every frame sent to the host CPU, or only into dedicated management frames (BPDUs).

In addition to BPDU frames, two types (PRIORITY_TYPE1/2) can be defined that will cause the tag being inserted if it should not be done on all frames. When using this feature, the type field compare can be done only for the first type field in the frame (i.e. it cannot skip VLAN tags at this point in the datapath where the decision for the tag insertion is done).

When the CPU sends a frame to the switch, it can insert the control information as needed, every frame can be treated individually and software can decide for the best implementation on a frame by frame basis.

When using the management tag features, the line ports must be configured to remove CRC from incoming frames (COMMAND_CONFIG_P[n].CRC_FWD = 0)

4.5.6 MAC EEE Support

4.5.6.1 Overview

Energy Efficient Ethernet (EEE) is defined in IEEE 802.3az. It defines energy saving modes for backplane applications allowing the PHY device to power down the link during idle times.

The MAC Core implements the necessary functionality to signal Low Power Idle (LPI) to the PHY.

The MAC includes the necessary timers and a control register to allow for the following usage scenarios:

1. Fully autonomous operation:
   The MAC integrated timers detect idle and initiate LPI transmission to power down the link when the idle time is reached. It also automatically wakes up and performs the Tw_sys_tx wake up delay before transmitting traffic.
2. Software controlled operation:
   A control register (EEE_CTL_STAT_P[n]) within the MAC allows controlling the transmission of LPI (lpi_req) and holding the transmitter during wakeup (lpi_txhold). This allows software to implement idle detection and wake up timers.

The following figure shows the implementation of the MAC Core.

Figure 4.21 MAC Extension for Autonomous EEE Operation

The following 3 functions are integrated with the MAC Core on every line port:

- A control/status register. Allows software to configure the mode of operation (full autonomous with timers, or software controlled) and inspect the received LPI indication status.
- An Idle timer. This timer will assert lpi_req when the MAC does not transmit any data for the configured amount of time. It will cause the RS layer to transmit LPI.
- A Wake up timer (Tw_sys_tx). This timer prevents the MAC transmitter from transmitting traffic for a certain period after wakeup. It will keep the lpi_txhold signal asserted after lpi_req deasserted.

All timers are programmable through registers under software control.

4.5.6.2 PHY Interface Encodings for EEE

The MAC general purpose PHY interface allows interconnecting to any PHY interface using a PHY interface converter as needed. The following tables list the encodings for standard GMII and MII to allow transferring the Low Power Idle (LPI) sequences between MAC and PHY.

(1) GMII Encodings for EEE

The following table shows the interface encodings. When no frame is transferred on the interface, specific idle sequencing can be encoded on the GMII asserting the error signal while data valid is kept low.

Table 4.235 GMII Encodings for EEE

rxdv/txen	rxer/txer	rxd/txd(7:0)	Description
1	x	xxh	Normal frame transfer
0	0	xxh	Normal interpacket gap (idle)
0	1	01h	Low Power Idle indication
0	1	0Fh / 1Fh	Carrier Extend / Error during half-duplex operation only (not supported)
0	1	others	Reserved

(2) MII Encodings for EEE

The following table shows the interface encodings. When no frame is transferred on the interface, specific idle sequencing can be encoded on the MII asserting the error signal while data valid is kept low.

Table 4.236 MII Encodings for EEE

rxdv/txen	rxer/txer	rxd/txd(3:0)	Description
1	x	xh	Normal frame transfer
0	0	xh	Normal interpacket gap (idle)
0	1	1h	Low Power Idle indication
0	1	others	Reserved

4.5.7 MAC Transmit Rate Control

4.5.7.1 General

Every MAC implements a rate control function to allow reducing the average transmitted traffic load on the link. When enabled, the transmission bandwidth can be configured to use from 1% to 80% of the available link bandwidth. When disabled, 100% link bandwidth are used.

The implemented algorithm uses the credit based shaper as defined in **Section 4.6.2**, (3). The algorithm maintains a credit counter for the traffic being transmitted by the MAC. The counter is incremented whenever a frame is waiting for transmission, or if the current count is negative. It decrements when a frame is transmitted. A frame can be transmitted only when the credit counter has a value of ≥ 0. If no frame is waiting and the credit counter is > 0, it is reset to 0.

The increment rate of the counter is defined by the idle slope. The decrement rate of the counter is defined by the send slope. The idle slope is configurable to define the bandwidth that can be allocated for the traffic. The send slope is defined by the link speed.

4.5.7.2 Configuration Settings

The send slope depends on the transmission speed of the link (e.g. 10 or 100 or 1000 Mbps) and is defined by the current mode of operation of the MAC. The idle slope is defined by configuration (see **Section 4.4.91, IDLE_SLOPE_P[n] — PORT[n] MAC Traffic Shaper Bandwidth Control (n = 0..4)**) depending on the (relative) bandwidth that should be allocated for the traffic.

The idle slope can be configured with a granularity of 1/64 bit. That is, the credit counter increments by 1 byte when the increment has reached 512 (8 × 64). The implementation may have restrictions on the granularity depending on byte to clock ratio of the interface, however the calculations will be based always on the 1/64-bit steps. This means the idle slope allows to specify bandwidth usage in steps of 0.2% (1/(1+512)) of the link capacity. Using an 11-bit value allows to configure the bandwidth allocation from 0.2% to 80%.

The resulting Bandwidth is calculated as:

$$\text{bandwidth [\%]} = 1 / (1 + 64/\text{IdleSlope} \times 8) = 1 / (1 + 512/\text{IdleSlope})$$

$$\text{IdleSlope} = (512 \times \text{bandwidth}) / (1 - \text{bandwidth})$$

Table 4.237 Credit Based Shaper Bandwidth Setting Examples

IdleSlope Value	Allocated Bandwidth	Notes
2	1/(1+32×8) = 0.4%	Counter decrements 8-bit per byte transmitted. During idle, it increments 2/64 bit per byte clock → requires 32×8 bytes clock cycles to increment by 1 byte.
4	1/(1+16×8) = 0.77%	During idle, it increments 4/64 bit per byte clock → requires 16×8 bytes clock cycles.
6	1/(1+10.6×8) = 1.1%	During idle, it increments 6/64 bit per byte clock → requires 10.6×8 bytes clock cycles.
64	1/(1+8) = 11%	Counter decrements 8-bit per transmit clock cycle. During idle, it increments 1 bit per transmit clock cycle hence requires 8 clock cycles.
57	1/10 = 10%	
73	1/8 = 12.5%	
170	1/(1+3) = 25%	
256	1/(1+0.25×8) = 33%	
512	1/(1+1) = 50%	
1536	1/(1+0.04×8) = 75%	
2046	80%	

4.5.8 Port Based Access Control (802.1X)

4.5.8.1 General

The IEEE 802.1X specification defines procedures to allow a port of the switch to be removed from the network until it is authorized by a higher layer authentication function.

Use of IEEE 802.1X is fully controlled by the application (software) usually connected through the management port. The application exchanges authentication messages through the switch with the devices that connect to the switch line ports. The switch does not interpret the messages aside from detecting the authentication relevant frames (EAPOL) and restricting forwarding if a port is configured as unauthorized.

Authentication of a port is controlled for each port individually through its configuration register AUTH_PORT[n].

4.5.8.2 Terms & Definitions

- PAE:
  Port Access Entity; An entity associated with a port that implements the authentication protocol(s) of interest.
- EAP:
  Extensible Authentication Protocol; Messages for encapsulation in frames for exchange between PAEs.
- EAPOL:
  EAP Over LAN; Encapsulation of EAP frames in IEEE 802.3 is defined as IEEE 802.1X:
  - Destination MAC address is either the PAE group address 01-80-c2-00-00-03 or the port's defined unicast address configured in the MAC of the port.
  - The ether type is set to the PAE Ethernet Type: 888eh
  - The frame shall not be VLAN tagged (VID != 0) but may be priority tagged.
- Port Controlled Direction. A port, while unauthorized, can operate with controlled direction "in" or "both":
  - When using controlled direction "in", the port only restricts incoming traffic but allows other ports to forward traffic to it for transmission. In controlled direction mode "both", forwarding traffic from other ports is not allowed.
  - An exception from this rule is the dedicated EAPOL management port, which must be allowed to forward EAPOL (or other) frames even to ports that have controlled direction "both" to implement the necessary authentication protocol.

4.5.8.3 Functions

When a port is configured as unauthorized, the MAC/switch port receive behavior changes as follows:

- Accept EAPOL frames and forward them exclusively to a defined port (so-called EAPOL port, which is typically the management port).
- Non EAPOL frames can either be discarded or be allowed for limited forwarding to a specific group of ports (guest VLAN). e.g. allow DHCP before authentication.
- Allow or discard non EAPOL frames received from other ports to be forwarded for transmission on the port (Controlled Directions "in" or "both").
- Allow or discard BPDU frames

When a port is configured as authorized, normal destination lookup and forwarding resolution occurs with additional filtering of unauthorized destination ports which are configured in controlled direction "both".

4.5.8.4 Forwarding Decision

The following decision is implemented in each port receive datapath to determine the destination port(s) of a frame or discard it depending on authentication status of a port.

Figure 4.22 IEEE 802.1X Forwarding Decision Flow

4.5.8.5 Usage aspects

The following points should be considered when implementing the authorization application:

- Source address learning should be disabled for an unauthorized port (see **Section 4.4.8, INPUT_LEARN_BLOCK — Input Learning Block Register**) to avoid invalid addresses can enter or overwrite the MAC address table (MAC address spoofing). The authorization application will need to enter the address into the MAC address table once the remote device is authenticated.
- With learning disabled and guest traffic allowed for an unauthorized port, the source address will not be entered into the address table of the switch. Hence all traffic sent to such an unauthorized port from another (answering) port is flooded. Flooding is limited to either the guest port mask of the (answering) port if it is unauthorized or the usual U/M/BCast flooding masks if the port is authorized. An exception is the management port, which can use forced forwarding to direct the frame to the intended port only.
- To enable EAPOL reception, the management port functionality must be enabled. That is register MGMT_CONFIG bit 6 (enable) must be 1 and bit 7 (discard) must be 0. Otherwise frames are discarded before EAPOL checking can be done.
- If a port is unauthorized, its cut through bit should be cleared to avoid any invalid frames can enter the switch. Cut Through should be enabled only on authorized ports.

4.5.9 Interrupts

The following interrupts are available and can be individually enabled through the INT_CONFIG register.

The INT_STAT_ACK register allows inspecting interrupt status and acknowledging the individual pending interrupts by writing a bit with “1” to clear the interrupt latch.

Table 4.238 Interrupt Sources

Source	Bit symbol of INT_CONFIG register	Description	Individual Control and Status/Ack register	Individual interrupt signal
MDIO	MDIO1	MDIO master transaction completion	—	—
Lookup Source	LK_NEW_SRC	A frame with an unknown source MAC address was received.	—	—
PHY Link Change	IRQ_LINK[3:0]	Change on any of the PHY link status	—	—
TX Timestamps	IRQ_TSM_TX[3:0]	Per Line Port indication that outgoing frame timestamp is available.	—	—
MAC EEE	IRQ_MAC_EEE[3:0]	The MAC integrated functions for managing Energy-Efficient Ethernet (EEE) can trigger interrupts from changing power states.	—	—
DLR Module	DLR_INT	Events generated by the Device Level Ring (DLR) module	DLR_IRQ_CONTROL DLR_IRQ_STAT_ACK	A5PSW_DLR_Int
PRP Module	PRP_INT	Events generated by the redundancy network support functions (PRP).	PRP_IRQ_CONTROL PRP_IRQ_STAT_ACK	A5PSW_PRP_Int
Hub Module	HUB_INT	Events generated by the Hub module.	HUB_IRQ_CONTROL HUB_IRQ_STAT_ACK	A5PSW_HUB_Int
Pattern Matcher Module	PATTERN_INT	Events generated by the Pattern Matcher module.	PTN_IRQ_CONTROL PTN_IRQ_STAT_ACK	A5PSW_PTRN_Int
TDMA Scheduler Module	TDMA_INT	Events generated by the TDMA Scheduler module.	TDMA_IRQ_CONTROL TDMA_IRQ_STAT_ACK	—

4.5.10 PHY Management Interface (MDIO Master)

4.5.10.1 Overview

The MDIO management interface is a two wire Management Interface. It provides a standardized method to access PHY device internal management registers. The IEEE 802.3 Clause 22 defines the bus protocol.

4.5.10.2 MDIO Frame Format

A complete frame has a length of 64 bits (32-bit preamble, 14-bit command, 2-bit bus direction change, 16-bit data). Each bit is transferred with the rising edge of the MDIO clock (MDC signal).

Table 4.239 MDIO Frame Formats (Read / Write)

Type	PRE	Command				TA	Data MSB LSB	Idle
		ST MSB LSB	OP MSB LSB	Addr1 MSB.LSB	Addr2 MSB.LSB			
Read	1 … 1	01b	10b	xxxxxb	xxxxxb	Z0	xxxxxxxxxxxxxxxb	Z
Write	1 … 1	01b	01b	xxxxxb	xxxxxb	10	xxxxxxxxxxxxxxxb	Z

Table 4.240 MDIO Frame Fields Description

Name	Description
PRE	Preamble: 32 bits of logical “1” sent prior to every transaction
ST	Start indication: “01b”
OP	The opcode defines whether a read or write operation is performed: 1. If “10b”, a read operation is performed. 2. If “01b”, a write operation is performed.
Addr1	The PHY device address (PHYADDR). Up to 32 devices can be addressed.
Addr2	Register Address. Each PHY can implement up to 32 registers.
TA	Turnaround time. Two bit-times are reserved for read operations to switch the data bus from write to read for read operations. The PHY device will present its register contents in the data phase and drives the bus from the 2nd bit of the turnaround phase.
Data	16 bits of data written to the PHY or read from the PHY.
Idle	Between frames, the MDIO data signal is tri-stated.

4.5.10.3 Turnaround Signalling

For Read operations, the addressed PHY has to drive the data bus. This direction change is done during the two bit-times TA phase of the frame.

Figure 4.23 MDIO Turnaround Sequence

When the last bit before the TA phase (<R> in above figure) has been transmitted to the PHY, the transmitter will switch its MDIO data signal into tristate (<Z>). One clock cycle later, the PHY starts driving the MDIO data signal to 0 (<0>) and then shifts out its 16 bits of data from the addressed register. After all 16 data bits have been transferred by the PHY, it again tri-states MDIO.

4.5.10.4 MDIO Control Registers

The configuration register MDIO_CFG_STATUS is used to set the operation parameters for all MDIO transactions and must be initialized before any transactions can occur.

The control registers are located at offsets 0700h.

4.5.10.5 Transaction Types

The MDIO provides two transactions. A read transaction and a write transaction. The transactions contain the PHY device address and the register address allowing to address up to 32 devices on the bus with up to 32 registers.

The transactions can occur after the MDIO_CFG_STATUS has been initialized.

(1) Read transaction

To read a register from the PHY the following sequence is used:

1. Setup MDIO_COMMAND with the PHY device and register address of interest and set bit 15 to trigger the read transaction.
2. Wait for transaction complete (inspecting MDIO_CFG_STATUS)
3. Read data from MDIO_DATA

(2) Write Transaction

To write into a PHY register, the following sequence is used:

1. Setup MDIO_COMMAND with the PHY device and register address of interest.
2. Write data into MDIO_DATA. This triggers the write transaction.
3. Wait for transaction complete (inspecting MDIO_CFG_STATUS)

4.5.11 DLR Extension Module

4.5.11.1 Overview

The A5PSW Device Level Ring (DLR) extension offers Beacon frame processing on receive for the two line-ports (0 and 1) of the switch implementation.

The DLR extension module operates between the MAC receive and switch input of a port. Beacon frames are detected and the frame parameters are interpreted and stored locally for software access.

The switch forwards the beacon frames normally (possibly using cut through if available). Using a static address entry in the MAC address table allows limiting beacon frame forwarding to the two DLR enabled ports only. This prevents any beacon traffic sent to the management port or other line ports of the switch avoiding unnecessary processing load.

In addition, a loop filter is inserted into each receive datapath to detect if the incoming frame's source address equals the local station MAC address. Such frames are discarded to avoid cycling of frames sent by the local device into the ring during reconfiguration times. The local node's MAC address can be configured with the DLR_LOC_MAClo/hi registers.

Any ring status change for beacon based node implementation is notified to the processor through the interrupt status. It is possible to read the received beacon frame parameters at any time. Statistics counters are implemented counting beacon frames processed.

Figure 4.24 Switch with DLR Extension

4.5.11.2 Beacon Frame Format

Within a DLR Network, the active ring supervisor transmits beacon frames through both of its Ethernet ports once per beacon interval (400 microseconds by default). DLR frames are using the frame format of 802.1Q. Frames are transmitted with highest priority (7). Beacon frames are DLR frames with a length of 64 bytes, excluding the preamble and the SFD bytes. Beacon frame consists of the following fields:

- Seven bytes preamble
- Start frame delimiter (SFD)
- Six bytes Destination MAC address of 01-21-6C-00-00-01
- Six bytes Source MAC address
- TPID (0x8100) and VLAN Info
- Ring Ether Type of 0x80E1
- Ring Subtype 0x02
- Ring protocol version 0x01
- Frame type 0x01
- Source port 0x00
- Source IP address
- Sequence ID
- Ring state
- Supervisor precedence
- Beacon interval
- Beacon timeout
- Frame check sequence (CRC value)

Table 4.241 Beacon Frame Format

Frame Length	Common DLR Protocol Header	7 octets	PREAMBLE
		1 octet	SFD
		6 octets	DESTINATION ADDRESS
		6 octets	SOURCE ADDRESS
		2 octets	TPID (0x8100)
		2 octets	VLAN info (0xE000 + VLAN ID)
		2 octets	Ring Ether TYPE (0x80E1)
		1 octet	Ring Sub Type(0x02)
		1 octet	Ring Protocol Version (0x01)
		1 octet	Frame Type (0x01)
		1 octet	Source Port (0x00)
		4 octets	Source IP Address (0x00 if source has no IP address)
		4 octets	Sequence ID
		1 octet	Ring State
		1 octet	Supervisor precedence
		4 octets	Beacon Interval
		4 octets	Beacon Timeout in microseconds
		20 octets	Reserved
		4 octets	Frame Check Sequence

Beacon frames are processed and parameters are stored locally for the software access. The following table shows the beacon frame fields.

Table 4.242 Beacon Frame Field Definitions

Term	Description
Destination Address	The destination MAC address for the Beacon frame is fixed multicast address of 01-21-6C-00-00-01. This is an exclusive MAC address used only for Beacon frames. Hence cut-through forwarding could be used based on this address match.
Source Address	Source MAC address of the supervisor. 48-bit address is stored in two separate registers.
TPID	DLR messages contain 2 octets TPID (0x8100) after the source MAC address according to IEEE 802.1Q.
VLAN info	16-bit information field contains the priority field and the VLAN ID. VLAN ID is configured at the ring supervisor and received by the ring nodes. The default value for the VLAN ID is 0 when there is no VLAN ID available. The default VLAN ID does not need to be changed unless a commercial switch is being used within the ring.
Ring Ether Type	Ether type for DLR frame is 0x80E1
Ring Sub Type	Ring Sub type value is always 0x02 for the DLR messages
Ring Protocol Version	Protocol version for the DLR messages
Frame Type	Frame Type value for the Beacon Frame is always 0x01
Source Port	Source port value is always 0x0 for the Beacon frame
Source IP address	IP address of the Supervisor. The default value for the IP address is 0 when there is no IP address available.
Sequence ID	Sequence identification number of the frame
Ring State	State of the Ring network transmitted by the ring supervisor.
Supervisor precedence	The Ring Supervisor Precedence value contains the user assigned precedence value to the ring supervisor. When multiple supervisors are enabled, the precedence value enables the user to select a single supervisor with highest precedence. The ring supervisor's precedence value can be any value within a range 0–255, with numerically higher values indicating higher precedence.
Beacon Interval	Interval in microsecond at which the ring supervisor sends Beacon frames. Valid value is in a range between minimum 100 microseconds and maximum 100 milliseconds. Typical value is 400 microseconds.
Beacon Timeout	Amount of time in microsecond, nodes shall wait before timing out reception of Beacon frames and taking appropriate action. Valid value is in a range between minimum 200 microseconds and maximum 500 milliseconds. Typical value is 1960 microseconds.
Frame Check Sequence	CRC value for the frame

4.5.11.3 Ring Node Functional Description

Beacon frames are detected and interpreted by the hardware so that the processor is not burdened with processing the Beacon frames for ring node implementation. Any ring status change for beacon based node implementation is notified to the processor through the interrupt status. It is also possible to read the received beacon frame parameters at any time. Statistics counters are also implemented to view the number of beacon frames processed.

(1) Configuration

Following are the steps to configure the DLR Module:

- Enter the Beacon destination address (01-21-6C-00-00-01) into the MAC address table as a static entry having only bits 0,1 set in the portmask (see also **Section 4.5.3.8, Layer 2 Look Up Engine Operational Description**). This ensures beacon frames are forwarded only between the two DLR capable ports within the switch.
- Set the Beacon destination address (01-21-6C-00-00-01) within the DLR module to allow its detection (see also **Section 4.5.11.3(2), Beacon Frame Detection**):
  - Set register DLR_DSTlo = 0x006c2101 which is also the reset value.
  - Set register DLR_DSThi = 0x0100 which is also the reset value.
- Set the MAC address of the local device which is used by the loop filter.
  - Set the first 4 bytes of the MAC address in register DLR_LOC_MAClo.
  - Set the last 2 bytes of the MAC address in register DLR_LOC_MAChi.
- Set the DLR Ethernet type value of 80E1h to the register DLR_ETH_TYP which is also the reset value.
- Enable and control the DLR module through the DLR_CONTROL register. It also defines the timer constant (cycles required to count 1 µs) used by the timeout counters.
- Configure the destination address for Neighbor_Check_xxx and Sign_On messages (01-21-6C-00-00-02) to ensure it is always forwarded to the management port only and not forwarded between any of the line ports. There are 2 alternatives to achieve this:
  - Configure the address into the additional BPDU register MGMT_ADDR0_lo/hi.
  - Add it as static address into the address table with the portmask having only the bit of the management port set to 1.
- Any interrupt bit can be enabled/ disabled through the register DLR_IRQ_CONTROL as required by the software.

(2) Beacon Frame Detection

Received frames are inspected by the DLR module only if they match the following rules. Otherwise the frame is ignored by the DLR module.

- Destination address must match the address provided in DLR_DSTlo/hi registers.
- Ethernet type must match the value given in register DLR_ETH_TYP.

If above rules both match, learning of the source address of the frame is suppressed, even if the port has learning enabled (to avoid the address table is constantly updated with alternating entries for the ring supervisor).

The accepted DLR frame is then forwarded to the beacon frame processing function (within the DLR module), which inspects the frame payload further to extract beacon information. However, this function accepts a frame only if the following rule matches:

- The DLR payload Frame Type field (3rd byte in payload) must have a value of 0x01.

If the DLR payload Frame Type is not matching, the frame is ignored by the beacon processing functions. However, source address learning will still not occur (as it does not depend on payload).

The switch will always process the frame normally (i.e. forwards it between line ports), independent of the DLR module function inspecting the frame.

(3) Start Up

Upon start up ring node will be in IDLE_STATE and presumes the network in linear topology mode. The current state of the local ring node and other status bits are stored within the register DLR_STATUS for software access.

Beacon frame received with invalid timer value will be ignored if ignore enabled through the DLR_CONTROL register. Invalid timeout timer value of the beacon frame will be always stored within the register DLR_INV_TMOUT irrespective of the ignore enable bit within the DLR_CONTROL register. An interrupt will be generated upon receiving a Beacon frame with invalid timeout timer value if enabled through DLR_IRQ_CONTROL.

Upon receiving a Beacon frame through either port, the node shall transition to FAULT_STATE, which presumes the network in ring topology mode. An interrupt will be generated to the processor indicating a need for flushing the MAC address learning table and a change in state if enabled through the register DLR_IRQ_CONTROL. All ring supervisor parameters will be saved within register for software access. Following parameters will be saved only during transition from IDLE_STATE to FAULT_STATE:

- Supervisor’s MAC address and will be stored in the register DLR_SUPR_MAClo and DLR_SUPR_MAChi
- Supervisor’s precedence value will be stored in the register DLR_SUPR_MAChi
- VLAN ID will be stored in the register DLR_STATE_VLAN
- Beacon timeout timer value will be stored in the register DLR_BEC_TMOUT

Supervisors IP address is accepted to change at any time. New IP address will always replace the old one and interrupt will be generated indicating the IP address change if enabled through the DLR_IRQ_CONTROL.

If beacon frame received from a different supervisor with a higher precedence value or higher numeric value for MAC address with same precedence value, new Beacon frame parameters will replace all the old values. An interrupt will be generated indicating change of supervisor if enabled through the register DLR_IRQ_CONTROL. Node will stay in FAULT_STATE.

If beacon frame received from a different supervisor with a lower precedence value or lower numeric value for MAC address with same precedence value, Beacon frame will be ignored. An interrupt will be generated indicating ignored beacon frame if enabled through the register DLR_IRQ_CONTROL. Node will stay in FAULT_STATE.

Ring supervisor is not expected to change parameters within the Beacon frame. If parameters need to be changed, supervisor is expected to stop transmitting Beacon frame for at least two beacon timeout periods before transmitting beacon frames with new parameters.

If the local node state changes back to IDLE_STATE due to beacon timeout timer expiration on both ports, it will generate interrupt if enabled through the register DLR_IRQ_CONTROL. Current interrupt status is available for software access showing beacon timeout timer asserted for both ports, need to flush MAC address learning table and state change within interrupt status register DLR_IRQ_STAT_ACK.

Upon receiving Beacon frame through both ports and after receiving a beacon frame from active ring supervisor with ring state field set to RING_NORMAL_STATE on either one of its ports, Local node shall transition to NORMAL_STATE. Interrupt status bits will show the change in state, need to flush the unicast MAC address learning table and to stop neighbor check timeout timer when active within the software.

NOTE

Neighbor check time out timer (100 millisecond) for neighbor check process (see **Section 4.6.2, References**, (5)) should be implemented by the software. Software may take advantage of the stop neighbor check timeout timer bit of the interrupt status register to stop the timer.

(4) Fault Detection

One of several possible events shall cause the ring node to transition from NORMAL_STATE to other states:

- Receipt of Beacon frame with state parameter set to RING_FAULT_STATE. Interrupt status will show a change in node state change and cause an interrupt if enabled
- Receipt of Beacon frame with different MAC address and higher precedence than the current active ring supervisor. In addition to state change, supervisor change bit will be asserted within the interrupt status register.
- Loss of beacon frame on both ports for a period specified by the Beacon timeout period will cause the node to transition to IDLE_STATE. In addition to the state change, Beacon timeout timer expire bits for both ports bit will be asserted within the interrupt status register.
- Loss of Beacon frame on a single port for a period specified by the Beacon timeout period will cause the node to transition to FAULT_STATE. In addition to the state change, Beacon timeout timer expire bits for corresponding port bit will be asserted within the interrupt status register.

(5) Error Handling

DLR node module is able to handle several error conditions,

- Beacon frames with CRC error are detected and not interpreted for DLR node implementation and discarded before entering the switch. Beacon frame parameters are not stored for the error frame. Beacon frames with CRC error are counted and stored in the DLR statistics register DLR_RX_ERR_STAT0 and DLR_RX_ERR_STAT1 for the port0 and port1 respectively.
- Valid range for the Beacon frame timeout timer value is between 200 microseconds to 500 milliseconds. Beacon frame from supervisor with invalid value of the beacon timeout will be ignored for processing (however are still forwarded normally by the switch) if configured through the DLR_CONTROL register. Beacon frames with invalid timeout timer are always detected irrespective of the DLR_CONTROL register configuration and invalid timeout value is stored within the register DLR_INV_TMOUT. An interrupt is also generated if enabled through the register DLR_IRQ_CONTROL.

(6) Beacon Frame Parameter Extraction

Parameters are transmitted by the Ring Supervisor through Beacon frame and extracted by the node and stored within read only Registers. There are three types of Beacon frame parameters depending on time of capture and store.

- Type1:
  Parameters which are not allowed to change during normal operation. This is done to avoid any unexpected change during normal operation. Parameters are always extracted and compared for the state machine evaluation but they are stored only during IDLE state or if the received Beacon frame has higher precedence than previously received Beacon frame.
- Type2:
  Allowed to change at any time. Such parameters are always extracted and stored if the Beacon frame is received from current Ring supervisor or node is in IDLE state or received Beacon frame has higher precedence than previously received Beacon frame.

- Type3:
  Always extracted from all Beacon frames on any port and stored within the register. Does not have any dependency on state or supervisor or precedence. Useful for debugging purpose only.

4.5.12 Parallel Redundancy Protocol (PRP)

4.5.12.1 Overview

The switch can be used within a redundant network infrastructure according to IEC 62439-3 Clause 4 (**Section 4.6.2**, (6)). The Parallel Redundancy Protocol (PRP) defines a network infrastructure consisting of two fully independent networks which are operated in parallel between all nodes. No communication occurs between the two networks and they are installed so that failure of one network does not affect the operation of the other network. Each network by itself is not aware of its role in a redundant infrastructure and is built from standard Ethernet networking equipment.

Each node that connects to the redundant network infrastructure uses two separate networking interfaces to connect to both networks at the same time. This is called a Double Attached Node (DAN). All DANs inject the same frame into both networks and append a Redundancy Control Trailer (RCT) which allows a receiving node to identify the redundant frames and only process one of the two frames. Hence if one of the networks experiences a failure the communication is not interrupted as all data still passes through the redundant network.

The infrastructure is held flexible to allow attachment of existing network components with only a single network interface (Single Attached Node, SAN) into the network. A SAN cannot benefit from the redundancy and can only communicate with other DAN or SAN on the same network.

If a SAN should also benefit from the redundant network infrastructure, it can be connected through a so-called Redundancy Box (RedBox) device which performs the necessary frame duplication functions to all traffic transferred between the SAN and the redundant networks. From the perspective of the network and all other nodes, a SAN behind a RedBox acts identical to a DAN and is called a virtual DAN (VDAN).

Figure 4.25 PRP Network Infrastructure Overview

4.5.12.2 A5PSW Function Summary

The A5PSW allows implementing a very flexible device by combining two of its ports to a “PRP Group” to connect to such a redundant network infrastructure. All traffic to and from the PRP grouped ports will be subject to redundancy processing like frame duplication, trailer append and duplicate detection.

At the same time, the other ports of the A5PSW can connect to regular SANs or other local (non redundant) networks and the switch will perform the function of a RedBox when forwarding traffic between the PRP Group and any other port. The following figure shows the different roles of the A5PSW within some mixed infrastructure example.

Figure 4.26 A5PSW within PRP Infrastructure

The implementation is held flexible and enables the following functionalities:

- Any two ports of the line ports can be grouped for redundancy (one group can exist).
- Perform frame duplication and RCT append to frames directed to double attached nodes (DANs) when transmitting to the PRP group ports.
- Perform duplicate detection on frames received from the redundant networks with optional dropping of duplicates.
- Avoid frame duplication and RCT append if the destination is known to be a single attached node (SAN).
- The switch forwarding function distinguishes DANs and SANs fully automatically using its lookup table. A DAN has both ports of the PRP Group listed as its destination, whereas a SAN only has one of the ports of the PRP Group as its destination.

- The learning function is enhanced to automatically detect DANs when it receives redundant traffic from a node through the PRP grouped ports.
- Alternatively, management can setup the forwarding table to define DANs and SANs.
- Several options exist to configure trailer append and duplicate detection functionalities with optional automatic removal of the RCT when forwarding to non redundant ports.
- The special local switch management port (internal port) which connects the local node to the network(s) allows to operate the necessary network monitoring (software) and management services and can act itself as a DAN attached to the redundant networks.
- Transparent RedBox functionality when forwarding traffic between the PRP grouped ports and the other switch ports.

4.5.12.3 Switch Forwarding Behavior

(1) Forwarding to Redundant Networks

Due to its flexible architecture and advantage of having a full switch in place, no dedicated redundancy layer exists. Instead, the existing switch forwarding mechanisms together with the capability to duplicate frames (used e.g. for broadcast or flooding) is used. When forwarding frames to the PRP grouped ports (as defined in register PRP_GROUP), the switch provides additional information with the frame to instruct the MAC transmitters to attach the redundancy trailer as needed. This allows transparent mixing of redundant and non redundant traffic on the networks. For example, link local traffic like BPDU management frames are forwarded without duplication and without redundancy trailer as are frames destined to SANs.

The following figure shows the flow of a frame and its respective actions along the path when forwarded from a non redundant port to the redundant ports. This effectively performs the RedBox functionality.

Figure 4.27 Forwarding to Redundant Ports (RedBox)

This also means that the switch's output queues are by itself already part of the redundant network infrastructure and sending of the duplicated frames into the redundant networks may not occur simultaneously. For example, if the queues of the redundant ports are filled to different levels because e.g. one port is transferring a frame to a SAN. Then the first of the two duplicates is sent immediately while the 2nd frame will have to wait for completion of the ongoing transmission at the other port.

To support automatic sequence number generation required to allow duplicate detection at the receiving DAN, the switch MAC address table is extended with a 2nd memory to store, for each MAC address, the current sequence number. The sequence number is automatically incremented whenever a frame from such a MAC address is forwarded from a non redundant port to the redundant ports and experiences duplication with trailer append. If the same source communicates with a SAN, the sequence number is not retrieved and stays unmodified.

(2) Receive from Redundant Networks

When receiving traffic from redundant networks optionally, a history memory is used to keep track of received sequence numbers. If the history memory shows that the same frame has been received from the same source already, it can be discarded. If it is the first of the two frames, it will be forwarded normally with the option to remove the redundancy control trailer.

Figure 4.28 Receiving from Redundant Networks

The figure shows an example where the duplicate is dropped in the upper path (from MAC LanA). The sequence numbers history memory is consulted by every port when it receives frames to identify duplicates. The switch will then mark the frame for discard and not forward it.

The history memory provides specific atomic functions that guarantee that simultaneous reception of frames will always generate a consistent dataset in memory that allows a decision to drop one of the redundant frames only.

If sequence numbers are unknown or out of the expected range, the history is capable to manage, duplicates will not be detected and both frames will be forwarded to the output queue(s). The higher layer protocols therefore will need to be robust against receiving duplicated frames, as there is no guarantee that duplicates will be dropped. Rather in case of doubt the forwarding of both frames tries to ensure that no data is lost.

4.5.12.4 MAC Address Learning Enhancements

The hardware MAC address table management allows automatic learning of source addresses when receiving frames (see also **Section 4.5.3.8(6), Automatic Learning / Migration**). For supporting redundant networks, it is enhanced to allow the same source address to appear on both redundant ports. That is, when the same MAC address is found in the source address field of frames received on both redundant ports, the associated port mask of the entry in the table will contain the same two bits that were defined as a PRP group (register PRP_GROUP).

This causes the switch forwarding to duplicate the frames to both redundant ports when a frame is sent to a node attached to both redundant networks (DAN). This also ensures that no duplication is performed if the address entry would not show both bits within its table identifying the remote device as a SAN.

In addition, dynamic entries indicate reception of frames from either of the two ports in the (otherwise unused) bits 51 and 52. The hardware will only set the corresponding bit when receiving frames from the designated LAN A or LAN B port (as defined by register PRP_GROUP). It is up to the management (software) to deliberately clear those bits as necessary. This can be used to e.g. verify that traffic from a DAN is received on both ports allowing detection of network failures in one of the redundant networks.

The following figure shows the enhanced format of dynamic entries when redundancy operation is enabled (see also **Figure 4.14, Address Memory Record Types**). The normally unused bits 51 and 52 are used to indicate reception from the corresponding port of a redundant network.

Figure 4.29 Dynamic Address Table Entry with Redundancy Enhancement

By disabling the learning function (register LK_CTRL), software can control the address table in a static way. However, it should then still use dynamic entries (i.e. Bit 49 = 0) when adding addresses for DANs to enable the automatic updating of Bits 51,52 for network monitoring purposes. The aging and migration functions can be disabled, hence dynamic entries will not be modified by the hardware except for updating (setting) those two bits of an entry.

CAUTION

The normal age bit (bit 50) is also set always by the hardware independent of the port where the frame was received from.

Table 4.243 PRP Support Relevant Registers

Register	Notes
PRP_GROUP	Defines which two of the line ports should be used to connect to redundant networks. The management port (internal Port 4 of A5PSW) cannot be assigned. When the switch forwards frames to the PRP group ports, it automatically performs frame duplication and appends the redundancy control trailer (RCT). A dedicated memory is used to store the current sequence number for each known (source) MAC address which is automatically incremented whenever such node sends frames to the redundant networks requiring duplication and trailer append. Frame duplication is controlled through the normal MAC address table entries (see **Section 4.5.3.7, Layer 2 Look Up Engine**). If an address is inserted into the table, the port bitmask can be set to contain both ports to identify a DAN. The MAC transmitters use this information to determine if a trailer should be appended or not. On the other hand, if only one port is listed, frames are forwarded as to any other port, no duplication occurs and no trailer is appended. ***Note*)** When using ports within a PRP group, the VLAN input and output manipulation function that relies on the per-VLAN tag mask cannot be used any more.
PRP_CONFIG	Enables all redundancy functions and configures when to add redundancy trailer or perform duplicate detection with automatic removal of the trailer.
PRP_IRQ_CONTROL	Allows enabling several interrupts that may be of interest to the management application when implementing network monitoring functionalities.
RM_ADDR_CTRL	Access functions to read and write the history (rx) and sequence number (tx) memory entries for every MAC address. The memory has the same number of entries as the MAC lookup table memory (LK_ADDR_CTRL) allowing to store a sequence number or sequence number history for each individual MAC address. Which type of information is stored depends on the port(s) associated with a MAC address: If it is associated with the redundant ports (i.e. it is a source address of a DAN), a history is stored allowing duplicate detection. If the MAC address is associated with any other port of the switch, a sequence number is stored which is used when appending the RCT to frames forwarded to a DAN. ***Note*)** The history and sequence number memory is automatically updated and managed by the switch hardware. The application usually does not modify the entries. It may however decide to e.g. clear the table or perform modifications during initialization or network reconfiguration scenarios.
LK_CTRL	The lookup function can be used to automatically learn addresses. This includes automatic detection of DAN and SAN within the redundant networks. Dynamic entries are enhanced to provide a (new) indication that allows if frames are received from a node through one or both of the redundant ports. Alternatively, management can set up the address table if the network configuration is known, or should be controlled by other measures. The management (software) then may choose to disable the global learning and aging functions (LK_CTRL bits 1,2,3) and can use the table management registers (LK_ADDR_CTRL) to add and remove addresses as needed. Even if using a static table setup, the entries for DANs and SANs should follow the format of dynamic entries (see **Figure 4.14, Address Memory Record Types**) as it allows the hardware to update the LAN A/B bits. The LAN A/B bits can be cleared by software and will be set by the switch hardware whenever the corresponding MAC source address is found in a frame received from any of the redundant ports. This method allows implementing a software controlled aging or additional monitoring for e.g. detecting faults within one of the networks connecting to a DAN.

4.5.13 Integrated HUB Module

4.5.13.1 Overview

The switch offers an integrated flexible Hub function. The Hub function emulates functionality similar to an IEEE 802.3 Clause 27 Class 2 repeater connecting multiple half-duplex ports to create a shared network infrastructure.

The basic task of the Hub function is to copy all frames in an immediate (< 460 ns) cut through manner between all ports. Only one port can receive at any time and data is always copied to all other ports, allowing for half-duplex operation only. If multiple ports receive simultaneously, a collision occurs resulting in corrupted data.

Figure 4.30 Integrated Hub Function Overview

The switch integrated Hub function allows flexible assignment of any line port to create a so-called Hub Group. The Hub Group implements a copying layer inserted between MAC and PHY interfaces with the following functionalities:

- Copying of data in between all ports of the Hub Group at the PHY interfaces.
- Copying of MAC transmit data from a single MAC of the group to all Hub Group PHY interfaces.
- Data received at PHY interfaces is transparently received by the corresponding MAC port (i.e. is not copied to multiple MAC receivers).

Additional switch functionalities that are available when the hub mode is active:

- Optional frame retransmission following collisions (when a MAC transmitted).
- Hub scheduler ensuring at all times only one MAC of the Hub group can transmit.
- Additional MAC transmit allowance controllable by application or from received frames (pattern matching).

4.5.13.2 Functional Description

The switch can forward (transmit) to the Hub Group with any of the MACs of the Hub Group. When a MAC transmits into the Hub Group, its data is copied to all PHY interfaces of the group. Data received at any of the Hub Group's PHY interfaces is received by the switch through the port's connected MAC layer as normal (i.e. no copying into multiple MAC receivers occurs).

The switch forwarding is modified to allow only one of the Hub Group's MACs to transmit at any time. The MACs of the Hub Group must be configured for half-duplex operation.

(1) Dataflow Example

The following figure shows an example Hub Group configuration consisting of ports 0 and 1.

Figure 4.31 Integrated Hub Copy Function Examples

(2) Transmit into Hub Group

When a switch port (port 0 in above example) within the Hub Group transmits, its MAC transmit data is copied to all group's PHY interfaces simultaneously. Note that any MAC of the group can transmit independently into the group

(and will be copied to all PHY interfaces). The hub scheduler function within the switch ensures that only one of the output ports can transmit at any time into the group to avoid collisions.

The ports which do not participate in the Hub Group are fully independent. Their PHY interfaces are directly connected to the corresponding MAC, bypassing the copy function, and may operate at different rates.

(3) Hub Group Receive

When a PHY within the Hub Group receives data, it is copied to all other PHY interfaces of the group, as well as to the corresponding MAC (i.e. PHY0 to MAC0, PHY1 to MAC1). The switch forwarding ensures that no frame received at a port of the Hub Group will be forwarded to the other ports of the Hub group to avoid frame duplication (and causing loops).

For learning, it means it will learn a MAC address from incoming frames at the port where it was received (same as normal switch mode). However, forwarding to a MAC learned on any of the Hub ports will still only use the dedicated (configured) Hub Group's default port.

4.5.13.3 Hub Specific Forwarding Rules

The switch forwarding decisions, when the Hub Group exists, needs special consideration for frames that are forwarded between the group and all other ports to avoid frame duplication. As every MAC within the Hub Group will transmit to all PHY interfaces of the group, frames would be sent multiple times if they are stored in multiple switch output ports (e.g. due to flooding or broadcast).

The switch forwarding decisions are modified to send frames always to only one MAC within the hub group (the default group port). When receiving frames from a port of the Hub group, the frame is never forwarded to any other port within the group (as this is already performed by the hub copy).

The following forwarding rules apply when the Hub is enabled:

- If frame received from within the Hub group
  - If isolate mode (HUB_CONFIG.HUB_ISOLATE = 1), set the management port as the only destination.
  - Otherwise, normal destination lookup defines the destination (incl. flooding) and all hub group ports are removed from final destination map.
- If frame received from the management port
  - If destination lookup fails or broadcast or BPDU: allow flooding only to the hub default port.
  - If destination lookup successful and a static entry: forward normally (can use static table entries to have full control to any hub group port, unfiltered!).
  - If destination lookup is successful and a dynamic entry:
    - If destination is a port of the hub group, forward to hub default port only;
    - Otherwise, forward normally (i.e. traffic to non hub ports).
  - Forced forwarding is unrestricted (application is responsible not to send to two ports within hub group).
- If frame received from another port (i.e. non hub group and not from management)
  - If destination lookup resolves non hub only, forward normally.
  - Otherwise, if destination is inside hub group (either from lookup or flooding),
    - If isolate mode: remove all hub group, add management port;
    - Otherwise, remove all hub group, add default hub port only.

The basic concept behind these rules is that all traffic from a port outside the Hub Group is sent only through the output port of a single MAC (the default hub port) into the hub infrastructure. Hence the other port(s) within the Hub group are not used for transmitting by default. They are however fully operable and the switch integrated hub scheduler will ensure that at any time only one of the MACs within the Hub Group will transmit into the hub infrastructure.

Those unused transmit MACs are available for use by the management port (application), which can use either forced forwarding or static address table entries to direct traffic to those ports.

For example, the default port would be used for all asynchronous traffic, while the management port can use the alternate port for all isochronous traffic and the protocol specific frames that require guaranteed response times.

4.5.13.4 HUB Group Clocking

When use Hub mode, set “RMII half-duplex 100 Mbps REF_CLK output” (RMII_100M_HALF_RO) mode to “RGMII/RMII Converter[m] Control register (m = 1..5)”.

4.5.13.5 PHY Requirement

A PHY that should be used with a port that is intended for use within a Hub Group requires the following functionality for proper operation when hub mode is active.

1. The PHY, when operating in half-duplex, must not return transmit data to its receive interface.
2. The carrier sense (CRS) signal must assert for any traffic activity (standard behavior).
3. Both (tx and rx) PHY interfaces (including the carrier sense signal) must be synchronous to a common clock domain (see **Section 4.5.13.4, HUB Group Clocking)**.
4. The PHY must operate in 100 Mbps at all times. Especially, when it has no link, it must be avoided that it would fall back to operate at 10 Mbps. A misconfigured speed on a port can prevent the hub from becoming operational hence affecting all ports of the group (see also **Section 4.4.191, HUB_STATUS — HUB Status Register**).

Note that the first two requirements are general requirements for any PHY that is used for half-duplex operation attached to any of the switch ports.

4.5.13.6 Usage Information

- The Hub Group can be enabled at any time, however due to the changing clocking requirements special care on possible reset sequences and PHY initialization may be needed.
- A Hub Group should not use ports used in other grouping modes at the same time (e.g. DLR, PRP grouped ports).
- All ports used within the Hub Group must be configured for 100 Mbps and half-duplex operation (COMMAND_CONFIG_P[n])

4.5.14 Receive Pattern Matcher

4.5.14.1 Functional Description

Every port implements additional receive pattern matchers which allow pattern search within up to 12 bytes following the frame's source address field.

The pattern compare starts at the 13th byte of a frame and can inspect up to the 24th byte (not considering Preamble/SFD). Hence it covers the length/type field of the frame and up to 10 bytes of payload (see also **Figure 4.2, MAC Frame Format Overview** for frame formats). For each byte, an individual compare value and bit mask are available to define individual length patterns.

A pattern defines a compare value (see registers PTRN_CMP*[1]) and a mask value (see registers PTRN_MSK*[2]). The mask value is ANDed with the frame's data and the result compared with the compare value to determine a match. Hence all bits that should be treated as not relevant must have a 0 in the mask as well as in the compare value itself.

The pattern matchers can be used in combination with the integrated hub module (see **Section 4.5.13, Integrated HUB Module**) to allow triggering a MAC transmission when a specific frame is received. This allows to implement protocols like where a node's transmission into the network is controlled by receiving specific frames from a central master node.

However, its use is not limited to the hub module but can be used for any applications.

Note 1. See **Section 4.4.198** to **Section 4.4.200**.
Note 2. See **Section 4.4.201** to **Section 4.4.203**.

4.5.14.2 Usage Information

- Up to 8 patterns are available, globally (see registers PTRN_CMP_xx*[1], PTRN_MSK_xx*[2] and PATTERN_SEL*[3]).
- The executed function if a pattern matches a received frame, is configured with the per pattern control register PATTERN_CTRL[n] (n=0..7).
- If and which pattern(s) is applied to traffic received at a specific port can be configured individually per port with the port specific RXMATCH_CONFIG[n] register. Multiple patterns can be enabled for a port implementing a logical OR (i.e. allowing any pattern to match and execute its corresponding function).
- For each pattern, an interrupt can be generated when a match occurs. This is globally per pattern, not per port, however still requires that a pattern is enabled at any one port at least. The pattern interrupts are enabled with the module's specific PTN_IRQ_CONTROL and PTN_IRQ_STAT_ACK registers.

Note 1. See **Section 4.4.198** to **Section 4.4.200**.
Note 2. See **Section 4.4.201** to **Section 4.4.203**.
Note 3. See **Section 4.4.197**.

4.5.15 TDMA Operation

4.5.15.1 Overview

To support network infrastructures that define a time multiplexed access to reserve bandwidth for different traffic classes, the switch can be configured to operate some or all ports in a TDMA (Time Division Multiple Access) fashion.

Typical usage of such a scheduling is to reserve bandwidth and define times (time slot) for transmitting e.g. real time or delay sensitive traffic. Other traffic can use the network only at other times. This allows implementation of isochronous (real time) traffic channels with deterministic delays while sharing the network with other best effort traffic.

Proper operation of such an infrastructure requires that all nodes and switches connected in the network operate with synchronized clocks (e.g. using IEEE 1588) and agree on the time slot assignments through some network management protocol.

4.5.15.2 Output Queue Scheduling

A group of ports (including all) can be assigned for operation in TDMA mode. The output queues of a port can be assigned freely to different time slots allowing a queue to transmit only within its granted time slot. The assignment of the time slots occurs periodically hence separating the available bandwidth to different network traffic classes stored in different priority queues of a port.

The TDMA scheduler allows assignment of 4 time slots within a periodically repeating time interval (cycle interval). For each time slot an individual set of queues, including none, can be allowed to transmit. Frames received and classified to a queue will immediately become transmitted as long as the queue has granted transmit time.

If multiple queues are enabled within a time slot, the normal priority resolution between the enabled queues occurs, serving the highest queue first (strict priority).

Using 4 time slots gives flexibility to allow different queues at different times, or define guard times between slots where no new transmission is allowed to begin. A guard slot can guarantee that all transmit and receive operations of a previous slot can finish before the next slot begins.

Note that receive is not restricted in any way when TDMA is active. Frames can arrive completely asynchronously at the switch but will be stored and sorted into time slots for transmission according to their classification (i.e. priority resolution).

4.5.15.3 Scheduling Example

The following figure shows the principle and definitions used when configuring the switch for TDMA operation. The example shows using two active time slots (duration: Ts–T1 and T2–T3) and two guard slots (duration: T1–T2 and T3–Ts+1) where no transmission is allowed to begin.

At every start of a cycle (Ts), the offsets for the slots are calculated from the settings T1, T2, T3. Then at all 4 times (Ts, T1, T2, T3) a different set of Queues is given transmit allowance.

In the example shown in the figure below, queue 3 is enabled from start of cycle (Ts) to T1, and the other queues 0,1,2 are enabled from T2 to T3. None of the queues is enabled during all other times. Hence the frames are scheduled as follows:

1. At cycle begin (Ts), frames stored in queue 3 are transmitted (e.g. data created by the local node during the previous cycle that could not be transmitted). As long as T1 is not reached received frames classified for queue 3 are immediately forwarded (e.g. orange frame 2).
2. At time T1, no more queues are allowed to transmit. Ongoing transmit and receive operations continue until they complete (i.e. 1 max. frame time).

3. At time T2, the queues 0,1,2 are enabled and due to priority of queue 1 over queue 0 the first frames are sent from queue 1. Finally, the frame from queue 0 is sent. Note that queue 2 had no data during this cycle but would have been scheduled before queue 1 otherwise.
4. At time T3, no more queues are allowed again. Ongoing transmit and receive operations are completing but no new transmit will begin.

The sequence repeats with queue 3 transmit allowance again when the next cycle begins at time Ts+1. If a queue has no data to transmit while it is granted time for transmission, the port stays quiet (i.e. reserved bandwidth is unused).

Figure 4.32 TDMA Scheduling Example

The TDMA Periodic Cycle Set register and each Time Offset registers define 4 values: Tcycle defines the periodic cycle, T1 defines the first offset from cycle start, T2 the second and T3 the last.

For the cycle start and each specified offset, any set of the available queues can be given the right to transmit (and those queues stay enabled until timer reaches the next offset boundary).

4.5.15.4 Startup Sequence

When the application has determined all information for the cycle and slots to use and the system timer has been locked to some master clock within the network infrastructure (and hence runs precise), it can start the TDMA scheduler.

To start the TDMA scheduler, the application defines the absolute time when to begin with the first cycle setting register TDMA_START. It then enables the scheduler writing register TDMA_CONFIG (bit0 = 1).

CAUTION

The application must ensure to write the TDMA_CONFIG before the value in TDMA_START can be reached by the timer. If it would enable the scheduler after the timer has already passed the start value, the scheduler will wait until the timer wraps around and reaches the value again one full timer period later, which may not be intended.

This will cause the TDMA scheduler to begin its operation as soon as the timer reached the configured start time. While waiting for the first cycle, a specific set of queues can be enabled with register QUEUES_START including none.

When the first cycle has started, the cycle interrupt occurs if enabled (see **Section 4.4.218, TDMA_IRQ_CONTROL — TDMA Interrupt Control Register**)

The TDMA scheduler can be disabled at any time by clearing the TDMA_CONFIG register. This has an immediate effect and will enable all queues again as for normal operation.

4.5.15.5 Usage Information

- When using the TDMA scheduler, the strict priority selection scheme must be configured (see **Section 4.4.37, IMC_CONFIG — Input Memory Controller Configuration Register**). Other schemes are not allowed and may give unpredictable results.
- The MAC rate limiter for all TDMA enabled ports must be disabled to avoid delaying transmissions unexpectedly (see **Section 4.4.91, IDLE_SLOPE_P[n] — PORT[n] MAC Traffic Shaper Bandwidth Control (n = 0..4)**).
- To classify received traffic into corresponding output queues, several methods can be used and combined (see also **Section 4.5.3.5(4), Priority Resolution**):
  - Classification based on settings per port with register PRIORITY_CFG[n].
  - Define mapping of VLAN priorities per port with register VLAN_PRIORITY[n].
  - Using the Pattern Matcher to define priority for a matching frame.

  As classification occurs at frame reception, all ports of the switch must be configured consistently, not only the TDMA enabled ports.
- The scheduler allows disabling time offsets T2 and T3 to allow implementing only 2 or 3 slots per cycle. Setting T2 to 0 will ignore T2 and T3 meaning the cycle is split into two slots with boundary at T1. Setting T3 to 0 will ignore T3 meaning the cycle is split into three slots with boundaries at T1, T2. T1 cannot be disabled.

4.5.16 Initializing A5PSW

The initializing sequence in this section is a example used for preparation of system environments for using A5PSW under configuration below.

Refer to “8.3.1 Initializing” in “the RZ/N1D Group, RZ/N1S Group, RZ/N1L Group User’s Manual: System Introduction, Multiplexing, Electrical and Mechanical Information” for Ethernet initialization.

Note 1. Section 10, Ethernet Accessory Register

Note 2. RZ/N1 User’s Manual: System Introduction, Multiplexing, Electrical and Mechanical Information

Note 3. **Section 4, Advanced 5port Switch (A5PSW)**

Figure 4.33 Initializing of A5PSW flowchart

4.6 Usage notes

4.6.1 Restriction

- If Ethernet ports are used as Hub mode, the ports must be configured to “RMII half-duplex 100 Mbps REF_CLK output” (RMII_100M_HALF_RO) mode by “RGMII/RMII Converter[m] Control register (CONVCTRL[m]) (m = 1..5)”. Because PHYs connected with the Ethernet ports in Hub mode requires common clock supplied as REF_CLK.
- To configure the port of A5PSW as half-duplex mode, it is necessary to clear PHY_DUPLEX bit of A5PSW Duplex Mode register (SWDUPC) and set HD_ENA bit of Port[n] Command Configuration register (COMMAND_CONFIG_P[n]).
- For initialization of the Learning Table, wait until Clear_Table bit of LK_CTRL register is cleared.
- IP Classify table isn’t initialized automatically, therefore it is necessary to initialize all areas by software. Refer to IP_PRIORITY[n] register description for an access procedure to IP Classify table.
- The PHY, when operating in half-duplex, must not return transmit data to its receive interface.
- TDMA scheduler may be stopped after a wrap-around under the specified condition, which the compare value after a full timer wraps around (modulo) to become less than the timestamp granularity. To avoid this, the configuration of time slots must consider the following rules. Also, use the Fine correction method for the timestamp:
  - Start of operation must be configured with an offset equal or larger the timestamp granularity of A5PSW_TS_NS_IN[31:0] which comes from GMAC1. A5PSW_TS_NS_IN is related to PTP modulation, and the timestamp granularity (= increment size) is decided by Sub_Second_Increment register of GMAC1. For the connection of A5PSW_TS_NS_IN, See *Section 8.2.2, Selection of clocks for PTP* in the *RZ/N1D Group, RZ/N1S Group, RZ/N1L Group User’s Manual: System Introduction, Multiplexing, Electrical and Mechanical Information*.
  - All TDMA scheduling time (T1, T2, and T3) must not to eliminate the workaround offset value.

Example)

Timestamp granularity:	20 ns (= Setting value of Sub_Second_Increment register of GMAC1)
Timer modulo:	1 s
Workaround offset:	20 ns (≥ timestamp granularity)

The register values for such set up would be:

TDMA_MODULO	= 1,000,000,000	(1 second timer modulo)
TDMA_CYCLE	= 20,000,000	(20 ms)
TDMA_T1	= 5,000,000	(5 ms)
TDMA_T2	= 15,000,000	(15 ms)
TDMA_T3	= 18,000,000	(18 ms)
TDMA_START	= 100,000,000 + 20	(100 ms + 20 ns; here the workaround offset is applied)

- Broadcast and flooding packets are forwarded between PRP ports as well.

  To avoid it, implement following measures

  - Disable packet forwarding to PRP ports by default mask registers (UCAST/MCAST/BCAST_DEFAULT_MASK)
  - Receive a TX packet by CPU through management port, then transmit the packet to PRP port by force forwarding function
- NodeTable function for RedBox is not implemented in this hardware. Please manage the NodeTable by software.
- When using ports within a PRP group, the VLAN input and output processing function cannot be used.
- When using Hub module, preamble of the frame output from A5PSW is extended compared with input frame (see the figure below). This is used to indicate the frame transmission to the next devices and to avoid the collision as soon as possible. From the Ethernet standard point of view, extension of preamble is permitted. However, it is found that some Ethernet POWERLINK support devices can't receive the frame with extended preamble. Therefore, do not use Hub module when supporting Ethernet POWERLINK.

Figure 4.34 Timing of Input and Output Frames in Case of Using Hub Module

- It is prohibited to use PRP Wrong ID LAN-A/-B Count Registers (CntErrWrongLanA and CntErrWrongLanB). Workaround by software is the following.

1. Mirror the incoming frames on PRP ports to the management port with management tag to know which port incoming frame comes from, LAN A or LAN B.
2. Host CPU confirms all the frames from management port whether LAN ID is correct or not by SW, and software counter is incremented if it is incorrect.

In order to reduce CPU load, mirroring frames can be reduced by setting MIRROR_CNT register.

Two options are available dependent on the situation as below, for example,

- Full inspection: For qualification you may want to be fully accurate, and mirror all the frames
- Production: Here sampling 1 out of N frames would be sufficient to detect errors. In this mode MIRROR_CNT can be set to N to reduce the load on the CPU.

Figure 4.35 Frame Forwarding by Mirroring Function

4.6.2 References

(1) RFC 2309; Recommendations on Queue Management and Congestion Avoidance in the Internet; IETF, 1998

(2) IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems; IEEE Std 1588-2008.

(3) IEEE 802.1Qav; Virtual Bridged Local Area Networks, Amendment 12: Forwarding and Queuing Enhancements for Time Sensitive Streams; 2009.

(4) IEEE 802.1X; Port Based Network Access Control; 2004.

(5) The CIP Networks Library, Volume 2, Ethernet/IP Adaptation of CIP. Edition 1.9, April 2009.

(6) IEC 62439-3; Industrial communication networks — High availability automation networks — Part 3: Parallel Redundancy Protocol (PRP) and High availability Seamless Redundancy (HSR); Edition 2.0; 2012 07.

Section 5 EtherCAT Slave Controller

5.1 Overview

Typical functions of EtherCAT Slave Controller and supported function are shown below. Regarding the detailed specification of EtherCAT and EtherCAT Slave Controller (ESC), refer to the documentation (e.g. ETG.1000 EtherCAT Specification) provided by EtherCAT Technology Group (ETG) and the EtherCAT Slave Controller IP Core (v2.04) datasheet provided by Beckhoff Automation GmbH.

Table 5.1 Typical Functions of EtherCAT Slave Controller and Supported Function (1/3)

Features	Functions	Support
EtherCAT protocol	Handling the following frames: • Ethernet frames with Ether type 88A4h • EtherCAT frames encapsulated in UDP/IP • EtherCAT frames with VLAN Tag • Normal Ethernet frames	✓
Addressing modes	Device addressing • Auto increment address • Configured station address • Broadcast address	✓
	Logical addressing	✓
Working counter	Counting the number of read/write from/to the device	✓
EtherCAT command type	Processing the command that master requests slaves to address each addressing mode	✓
Loop control	Loop control and loop state in ESC	✓
Shadow buffer	Shadow buffers function when register is read/written	✓
Circulating frames	Processing of circulating frames during the failure	✓
Link detection	Link MII signal (PHY link signal)	✓
	MI Link detection and configuration (monitoring the PHY register via the management interface)	—
	Enhanced link detection (monitoring the state of transfer by MII RX error monitor)	✓
FIFO size reduction	RX FIFO size reduction because of reduction of propagation delay	✓
Ethernet physical layer	MII	✓
	EBUS	—
	Back-to-Back MII connection	✓
	MII management interface	✓
	Read/write of the PHY register via MII management interface	✓
	PHY address offset	✓
	Manual TX clock shift compensation	✓
	Automatic TX clock shift compensation	✓
FMMU	Mapping between logical address and physical address	✓
SyncManager	Buffer mode	✓
	Mailbox mode	✓
	Interrupt and latch event generation when a buffer was completely and successfully written or read.	✓
	Repeating mailbox communication	✓
	SyncManager deactivation by the PDI	✓

Table 5.1 Typical Functions of EtherCAT Slave Controller and Supported Function (2/3)

Features	Functions	Support
Distributed clocks	Clock Synchronization considering propagation delay and drift compensation	✓
	Generation of synchronous output signals (SYNC0 and 1 signals) • Cyclic mode • Single shot mode • Cyclic acknowledge mode • Single shot acknowledge mode	✓
	Precise time stamping of input events (LATCH0 and 1 signals) • Single event mode • Continuous mode • SyncManager event mode (for debugging)	✓
	Generation of synchronous interrupts	✓
	Synchronous digital output updates / Synchronous digital input sampling	—
	Exclusive control for the SYNC and LATCH signals of the EtherCAT and PDI	✓
	System time control by the PDI	—
	Communication Timing • Free run • Synchronized to output event • Synchronized to SYNC signal	✓
EtherCAT state machine	Control of state machine / Indication of the status and error code	✓
	Device emulation	—
SII EEPROM	SII EEPROM commands	✓
	SII EEPROM error indication	✓
	SII EEPROM access interface	✓
	EEPROM size selection	✓
	EEPROM emulation	—
Interrupt	AL event request (PDI interrupt)	✓
	EtherCAT event request (EtherCAT interrupt)	✓
Watchdog	Process data watchdog	✓
	PDI watchdog	✓
Error counters	Port error counters	✓
	Forwarded RX error counter	✓
	EtherCAT processing unit error counter	✓
	PDI error counter	✓
	Lost link counter	✓
	Watchdog counter process data	✓
	Watchdog counter PDI	✓
LED signals	RUN LED signal	✓
	ERR LED signal	✓
	STATE LED and STATE_RUN LED signals	✓
	LINK/ACT LED signals	✓
	Port error LED signal	—
	RUN/ERR LED override	✓
Process data interface (PDI)	Digital I/O	—
	SPI slave	—
	8-bit/16-bit synchronous/asynchronous microcontroller interface	—
	On-chip bus	✓
	General purpose I/O	—

Table 5.1 Typical Functions of EtherCAT Slave Controller and Supported Function (3/3)

Features	Functions	Support
Write protection	Write protection for the register area (0000h to 0FFFh)	✓
	Write protection for the whole area including the user RAM and process data RAM (0000h to 2FFFh)	✓
ESC reset	ESC reset from the master or PDI	✓

5.1.1 Block Diagram

Figure 5.1 EtherCAT Slave Connections

5.1.2 Features

- Up to 3 ports
- Automatic TX Shift
- Enhanced Link Detection
- 8 FMMUs
- 8 SyncManagers
- 8 KByte Process Data RAM
- 64-bit Distributed Clocks
- Mapping to global IRQ
- Read/Write Offset
- Write Protection
- AL Status Code Register
- Extended Watchdog
- AL Event Mask Register
- Watchdog Counter
- SyncManager Event Times
- EPU (EtherCAT Processing Unit) and PDI Error Counter
- Lost Link Counter
- I2C interface for external EEPROM
- RESET slave by EtherCAT master or CPU
- RUN LED
- Extended RUN/ERR LED
- PLB v4.6 as PDI (converted to AHB by additional bridge module)
- Interface
  - Native mode MII
  - No Native RMII, managed by RMII/RGMII convertor (connected on external pins)

5.2 Signal Interface

Table 5.2 Signal Interface of the EtherCAT Slave Controller (excluding PHY MII pins)

Signal Name	I/O	Description	Active
Clock			
ECAT_HCLK	I	AHB clock	
ECAT_CLK100	I	100 MHz for EtherCAT	
ECAT_CLK25	I	25 MHz for EtherCAT	
Interrupt			
ETHCAT_RST_Int	O	ETHERCAT Reset interrupt, level sensitive	High
ETHCAT_SYNC_Int[0]	O	ETHERCAT Sync0 interrupt, pulse sensitive	High
ETHCAT_SYNC_Int[1]	O	ETHERCAT Sync1 interrupt, pulse sensitive	High
ETHCAT_WDT_Int	O	ETHERCAT WDT interrupt, pulse sensitive	High
ETHCAT_EOF_Int	O	ETHERCAT EOF interrupt, pulse sensitive	High
ETHCAT_SOF_Int	O	ETHERCAT SOF interrupt, pulse sensitive	High
ETHCAT_Int	O	ETHERCAT interrupt, level sensitive	High
External Signal			
CAT_LEDRUN	O	EtherCAT RUN LED port	High
CAT_LEDSTER	O	EtherCAT Dual-color State LED port	High
CAT_LEDERR	O	EtherCAT Error LED port	High
CAT_LINKACT0	O	EtherCAT link / Activity LED port (port A) (Port 0 of ESC)	High
CAT_LINKACT1	O	EtherCAT link / Activity LED port (port B) (Port 1 of ESC)	High
CAT_LINKACT2	O	EtherCAT link / Activity LED port (port C) (Port 2 of ESC)	High
CAT_SYNC0	O	EtherCAT SYNC0 port	High
CAT_SYNC1	O	EtherCAT SYNC1 port	High
CAT_LATCH0	I	EtherCAT LATCH0 port	Rise/Fall (Both Edge)
CAT_LATCH1	I	EtherCAT LATCH1 port	Rise/Fall (Both Edge)
CAT_MII_LINK[0]	I	EtherCAT PHY indicating a link (port A) (Port 0 of ESC)	High*[1]
CAT_MII_LINK[1]	I	EtherCAT PHY indicating a link (port B) (Port 1 of ESC)	High*[1]
CAT_MII_LINK[2]	I	EtherCAT PHY indicating a link (port C) (Port 2 of ESC)	High*[1]
CAT_RESETOUT_N	O	EtherCAT RESET OUT	Low
CAT_I2CCLK	O	EtherCAT EEPROM I2C clock port	—
CAT_I2CDATA	I/O	EtherCAT EEPROM I2C data port	—

Note 1. CAT_MII_LINK[2:0] active level is controlled by Ethernet PHY Link Mode register.

5.3 Register Map

Table 5.3 EtherCAT Register Map (1/3)

Address	Register Symbol	Register Name
4401 0000h	TYPE	Type register
4401 0001h	REVISION	Revision register
4401 0002h	BUILD	Build register
4401 0004h	FMMU_NUM	FMMUs supported register
4401 0005h	SYNC_MANAGER	SyncManagers supported register
4401 0006h	RAM_SIZE	RAM Size register
4401 0007h	PORT_DESC	Port Descriptor register
4401 0008h	FEATURE	ESC Features supported register
4401 0010h	STATION_ADR	Configured Station Address register
4401 0012h	STATION_ALIAS	Configured Station Alias register
4401 0020h	WR_REG_ENABLE	Write Register Enable register
4401 0021h	WR_REG_PROTECT	Write Register Protection register
4401 0030h	ESC_WR_ENABLE	ESC Write Enable register
4401 0031h	ESC_WR_PROTECT	ESC Write Protection register
4401 0040h	ESC_RESET_ECAT	ESC Reset EtherCAT register
4401 0041h	ESC_RESET_PDI	ESC Reset PDI register
4401 0100h	ESC_DL_CONTROL	ESC DL Control register
4401 0108h	PHYSICAL_RW_OFFSET	Physical Read/Write Offset register
4401 0110h	ESC_DL_STATUS	ESC DL Status register
4401 0120h	AL_CONTROL	AL Control register
4401 0130h	AL_STATUS	AL Status register
4401 0134h	AL_STATUS_CODE	AL Status Code register
4401 0138h	RUN_LED_OVERRIDE	RUN LED Override register
4401 0139h	ERR_LED_OVERRIDE	ERR LED Override register
4401 0140h	PDI_CONTROL	PDI Control register
4401 0141h	ESC_CONFIG	ESC Configuration register
4401 0150h	PDI_CONFIG	PDI Configuration register
4401 0151h	SYNC_LATCH_CONFIG	SYNC/LATCH PDI Configuration register
4401 0152h	EXT_PDI_CONFIG	Extended PDI Configuration register
4401 0200h	ECAT_EVENT_MASK	EtherCAT Event Mask register
4401 0204h	AL_EVENT_MASK	AL Event Mask register
4401 0210h	ECAT_EVENT_REQ	EtherCAT Event Request register
4401 0220h	AL_EVENT_REQ	AL Event Request register
4401 0300h + 2h × n	RX_ERR_COUNT[n] (n = 0..2)	Rx Error Counter [n] register
4401 0308h + 1h × n	FWD_RX_ERR_COUNT[n] (n = 0..2)	Forwarded Rx Error counter [n] register
4401 030Ch	ECAT_PROC_ERR_COUNT	EtherCAT Processing Unit Error Counter register
4401 030Dh	PDI_ERR_COUNT	PDI Error Counter register
4401 0310h + 1h × n	LOST_LINK_COUNT[n] (n = 0..2)	Lost Link Counter [n] register
4401 0400h	WD_DIVIDE	Watchdog Divider register
4401 0410h	WDT_PDI	Watchdog Time PDI register
4401 0420h	WDT_DATA	Watchdog Time Process Data register
4401 0440h	WDS_DATA	Watchdog Status Process Data register
4401 0442h	WDC_DATA	Watchdog Counter Process Data register
4401 0443h	WDC_PDI	Watchdog Counter PDI register

Table 5.3 EtherCAT Register Map (2/3)

Address	Register Symbol	Register Name
4401 0500h	EEP_CONF	EEPROM Configuration register
4401 0501h	EEP_STATE	EEPROM PDI Access State register
4401 0502h	EEP_CONT_STAT	EEPROM Control/Status register
4401 0504h	EEP_ADR	EEPROM Address register
4401 0508h	EEP_DATA	EEPROM Data register
4401 0510h	MII_CONT_STAT	MII Management Control/Status register
4401 0512h	PHY_ADR	PHY Address register
4401 0513h	PHY_REG_ADR	PHY Register Address register
4401 0514h	PHY_DATA	PHY Data register
4401 0516h	MII_ECAT_ACS_STAT	MII Management EtherCAT Access State register
4401 0517h	MII_PDI_ACS_STAT	MII Management PDI Access State register
4401 0518h + 1h × n	PHY_STATUS[n] (n = 0..2)	PHY Port Status [n] register
4401 0600h + 10h × n	FMMU[n]_L_START_ADR (n = 0..7)	FMMU Logical Start Address [n] register
4401 0604h + 10h × n	FMMU[n]_LEN (n = 0..7)	FMMU Length [n] register
4401 0606h + 10h × n	FMMU[n]_L_START_BIT (n = 0..7)	FMMU Logical Start bit [n] register
4401 0607h + 10h × n	FMMU[n]_L_STOP_BIT (n = 0..7)	FMMU Logical Stop bit [n] register
4401 0608h + 10h × n	FMMU[n]_P_START_ADR (n = 0..7)	FMMU Physical Start Address [n] register
4401 060Ah + 10h × n	FMMU[n]_P_START_BIT (n = 0..7)	FMMU Physical Start bit [n] register
4401 060Bh + 10h × n	FMMU[n]_TYPE (n = 0..7)	FMMU Type [n] register
4401 060Ch + 10h × n	FMMU[n]_ACT (n = 0..7)	FMMU Activate [n] register
4401 0800h + 8h × n	SM[n]_P_START_ADR (n = 0..7)	SyncManager Physical Start Address [n] register
4401 0802h + 8h × n	SM[n]_LEN (n = 0..7)	SyncManager Length [n] register
4401 0804h + 8h × n	SM[n]_CONTROL (n = 0..7)	SyncManager Control [n] register
4401 0805h + 8h × n	SM[n]_STATUS (n = 0..7)	SyncManager Status [n] register
4401 0806h + 8h × n	SM[n]_ACT (n = 0..7)	SyncManager Activate [n] register
4401 0807h + 8h × n	SM[n]_PDI_CONT (n = 0..7)	SyncManager PDI Control [n] register
4401 0900h	DC_RCV_TIME_PORT0	Receive Times Port0 register
4401 0904h	DC_RCV_TIME_PORT1	Receive Times Port1 register
4401 0908h	DC_RCV_TIME_PORT2	Receive Times Port2 register
4401 0910h	DC_SYS_TIME	System Time register
4401 0918h	DC_RCV_TIME_UNIT	Receive Time EtherCAT Processing Unit register
4401 0920h	DC_SYS_TIME_OFFSET	System Time Offset register
4401 0928h	DC_SYS_TIME_DELAY	System Time Delay register
4401 092Ch	DC_SYS_TIME_DIFF	System Time Difference register
4401 0930h	DC_SPEED_COUNT_START	Speed Counter Start register
4401 0932h	DC_SPEED_COUNT_DIFF	Speed Counter Diff register
4401 0934h	DC_SYS_TIME_DIFF_FIL_DEPTH	System Time Difference Filter Depth register
4401 0935h	DC_SPEED_COUNT_FIL_DEPTH	Speed Counter Filter Depth register
4401 0980h	DC_CYC_CONT	Cyclic Unit Control register
4401 0981h	DC_ACT	Activation register
4401 0982h	DC_PULSE_LEN	Pulse Length of SyncSignals register
4401 0984h	DC_ACT_STAT	Activation Status register
4401 098Eh	DC_SYNC0_STAT	SYNC0 Status register
4401 098Fh	DC_SYNC1_STAT	SYNC1 Status register
4401 0990h	DC_CYC_START_TIME	Start Time Cyclic Operation/Next SYNC0 Pulse register

Table 5.3 EtherCAT Register Map (3/3)

Address	Register Symbol	Register Name
4401 0998h	DC_NEXT_SYNC1_PULSE	Next SYNC1 Pulse register
4401 09A0h	DC_SYNC0_CYC_TIME	SYNC0 Cycle Time register
4401 09A4h	DC_SYNC1_CYC_TIME	SYNC1 Cycle Time register
4401 09A8h	DC_LATCH0_CONT	Latch0 Control register
4401 09A9h	DC_LATCH1_CONT	Latch1 Control register
4401 09AEh	DC_LATCH0_STAT	Latch0 Status register
4401 09AFh	DC_LATCH1_STAT	Latch1 Status register
4401 09B0h	DC_LATCH0_TIME_POS	Latch0 Time Positive Edge register
4401 09B8h	DC_LATCH0_TIME_NEG	Latch0 Time Negative Edge register
4401 09C0h	DC_LATCH1_TIME_POS	Latch1 Time Positive Edge register
4401 09C8h	DC_LATCH1_TIME_NEG	Latch1 Time Negative Edge register
4401 09F0h	DC_ECAT_CNG_EV_TIME	Buffer Change Event Time register
4401 09F8h	DC_PDI_START_EV_TIME	PDI Buffer Start Event Time register
4401 09FCh	DC_PDI_CNG_EV_TIME	PDI Buffer Change Event Time register
4401 0E00h	PRODUCT_ID	PRODUCT ID register
4401 0E08h	VENDOR_ID	Vendor ID register

Table 5.4 EtherCAT Memory Map

Address	Register Symbol	Register Name
4401 0F80h to 4401 0FFFh	USER_RAM	User RAM
4401 1000h to 4401 2FFFh	DATA_RAM	Process Data RAM

5.4 Register Description

5.4.1 TYPE — Type Register

This register indicates the type of the EtherCAT slave controller.

Address: 4401 0000h

Bit	b7	b6	b5	b4	b3	b2	b1	b0
	TYPE							
Value after reset	1	0	1	0	0	0	0	0

Table 5.5 TYPE Register Contents

Bit Position	Bit Name	Function	R/W
b7 to b0	TYPE	Type of the EtherCAT slave controller Accessing from EtherCAT master: Read only	R

5.4.2 REVISION — Revision Register

This register indicates the revision of the EtherCAT slave controller.

Address: 4401 0001h

Bit	b7	b6	b5	b4	b3	b2	b1	b0
	REV							
Value after reset	0	0	0	0	0	0	1	0

Table 5.6 REVISION Register Contents

Bit Position	Bit Name	Function	R/W
b7 to b0	REV	Revision of the EtherCAT slave controller Accessing from EtherCAT master: Read only	R

5.4.3 BUILD — Build Register

This register indicates the build number of the EtherCAT slave controller.

Address: 4401 0002h

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	BUILD															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 5.7 BUILD Register Contents

Bit Position	Bit Name	Function	R/W
b15 to b0	BUILD	Build number of the EtherCAT slave controller Accessing from EtherCAT master: Read only	R

5.4.4 FMMU_NUM — FMMUs Supported Register

This register indicates the number of FMMU channels supported in the EtherCAT slave controller.

Address: 4401 0004h

Bit	b7	b6	b5	b4	b3	b2	b1	b0
	NUMFMMU							
Value after reset	0	0	0	0	1	0	0	0

Table 5.8 FMMU_NUM Register Contents

Bit Position	Bit Name	Function	R/W
b7 to b0	NUMFMMU	Number of FMMU channels supported in the EtherCAT slave controller. Accessing from EtherCAT master: Read only	R

5.4.5 SYNC_MANAGER — SyncManagers Supported Register

This register indicates the number of SyncManager channels supported in the EtherCAT slave controller.

Address: 4401 0005h

Bit	b7	b6	b5	b4	b3	b2	b1	b0
	NUMSYNC							
Value after reset	0	0	0	0	1	0	0	0

Table 5.9 SYNC_MANAGER Register Contents

Bit Position	Bit Name	Function	R/W
b7 to b0	NUMSYNC	Number of SyncManager channels supported in the EtherCAT slave controller Accessing from EtherCAT master: Read only	R

5.4.6 RAM_SIZE — RAM Size Register

This register indicates the process data RAM size supported in the EtherCAT slave controller in Kbyte.

Address: 4401 0006h

Bit	b7	b6	b5	b4	b3	b2	b1	b0
	RAMSIZE							
Value after reset	0	0	0	0	1	0	0	0

Table 5.10 RAM_SIZE Register Contents

Bit Position	Bit Name	Function	R/W
b7 to b0	RAMSIZE	Process data RAM size supported in the EtherCAT slave controller (unit: Kbyte) Accessing from EtherCAT master: Read only	R

5.4.7 PORT_DESC — Port Descriptor Register

This register indicates the port configuration.

Address: 4401 0007h

Bit	b7	b6	b5	b4	b3	b2	b1	b0
	P3		P2		P1		P0	
Value after reset	0	0	1	1	1	1	1	1

Table 5.11 PORT_DESC Register Contents

Bit Position	Bit Name	Function	R/W
b7, b6	P3	Port 3 configuration: This LSI does not implement port 3. Fixed to (00b). 00b: Not implemented 01b: Not configured (SII EEPROM) 10b: EBUS 11b: MII Accessing from EtherCAT master: Read only	R
b5, b4	P2	Port 2 configuration: Fixed to the setting for MII connection (11b) in this LSI. 00b: Not implemented 01b: Not configured (SII EEPROM) 10b: EBUS 11b: MII Accessing from EtherCAT master: Read only	R
b3, b2	P1	Port 1 configuration: Fixed to the setting for MII connection (11b) in this LSI. 00b: Not implemented 01b: Not configured (SII EEPROM) 10b: EBUS 11b: MII Accessing from EtherCAT master: Read only	R
b1, b0	P0	Port 0 configuration: Fixed to the setting for MII connection (11b) in this LSI. 00b: Not implemented 01b: Not configured (SII EEPROM) 10b: EBUS 11b: MII Accessing from EtherCAT master: Read only	R

5.4.8 FEATURE — ESC Features Supported Register

Address: 4401 0008h

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	FSCONFIG	RWSUPP	LRW	DCSYNC	FCS	LINKDECMII	—	—	DCWID	DC	—	FMMU
Value after reset	X	X	X	X	0	0	0	1	1	1	X	X	1	1	X	0

Table 5.12 FEATURE Register Contents (1/2)

Bit Position	Bit Name	Function	R/W
b15 to b12	Reserved	When read, the value returned is undefined. Accessing from EtherCAT master: Read only	R
b11	FSCONFIG	Fixed FMMU/SyncManager configuration 0: Variable configuration 1: Fixed configuration Accessing from EtherCAT master: Read only	R
b10	RWSUPP	EtherCAT read/write command support (BRW, APRW, FPRW) 0: Supported 1: Not supported Accessing from EtherCAT master: Read only	R
b9	LRW	EtherCAT LRW command support 0: Supported 1: Not supported Accessing from EtherCAT master: Read only	R
b8	DCSYNC	Enhanced DC SYNC activation 0: Not available 1: Available Accessing from EtherCAT master: Read only	R
b7	FCS	Separate handling of FCS errors 0: Not supported 1: Supported. Frames with wrong FCS and additional nibble will be counted separately in forwarded RX error counter. Accessing from EtherCAT master: Read only	R
b6	LINKDECMII	Enhanced link detection in MII 0: Not available 1: Available Accessing from EtherCAT master: Read only	R
b5, b4	Reserved	When read, the value returned is undefined. Accessing from EtherCAT master: Read only	R
b3	DCWID	Distributed clocks (width) 0: 32 bits 1: 64 bits Accessing from EtherCAT master: Read only	R
b2	DC	Distributed clocks 0: Not available 1: Available Accessing from EtherCAT master: Read only	R
b1	Reserved	When read, the value returned is undefined. Accessing from EtherCAT master: Read only	R

Table 5.12 FEATURE Register Contents (2/2)

Bit Position	Bit Name	Function	R/W
b0	FMMU	FMMU operation 0: Bit oriented 1: Byte oriented Accessing from EtherCAT master: Read only	R

5.4.9 STATION_ADR — Configured Station Address Register

Address: 4401 0010h

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	NODADDR															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 5.13 STATION_ADR Register Contents

Bit Position	Bit Name	Function	R/W
b15 to b0	NODADDR	Address used for node addressing (FPxx commands) Accessing from EtherCAT master: Read only	R/W

5.4.10 STATION_ALIAS — Configured Station Alias Register

This register indicates the alias address used for node addressing (FPxx commands).

NOTE

The initial value, 0, is retained until the EEPROM is loaded. After that, the value becomes the value at address 0004h in the EEPROM. This value is only taken over from the EEPROM the first time the EEPROM is loaded after a power on or reset.

Address: 4401 0012h

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	NODALIADDR															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 5.14 STATION_ALIAS Register Contents

Bit Position	Bit Name	Function	R/W
b15 to b0	NODALIADDR	Alias address used for node addressing (FPxx commands). The use of this alias is activated by setting bit 24 of the ESC DL Control Register (ESC_DL_CONTROL at 0100h) to 1. Accessing from EtherCAT master: R/W	R

5.4.11 WR_REG_ENABLE — Write Register Enable Register

This register is used to unlock the write protection temporarily while registers are write protected.

Address: 4401 0020h

Bit	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	ENABLE
Value after reset	X	X	X	X	X	X	X	0

Table 5.15 WR_REG_ENABLE Register Contents

Bit Position	Bit Name	Function	R/W
b7 to b1	Reserved	When read, the value returned is undefined. When writing to these bits, write 0. Accessing from EtherCAT master: R/W	R
b0	ENABLE	When registers are currently being protected against writing (bit 0 is 1 in the write register protection register, WR_REG_PROTECT, at 0021h) and freely writing to registers of the given node is to be permitted, the operation to do so by writing to this register has to proceed in the same Ethernet frame and preceding the other desired writing to registers. Write protection will be reactivated once the frame period elapses (unless the value in the write register protection register is changed). Accessing from EtherCAT master: R/W	R

5.4.12 WR_REG_PROTECT — Write Register Protection Register

This register is used to protect registers against writing. The registers in the area 4401 0000h to 4401 0FFFh are write-protected (except for the WR_REG_ENABLE register (0020h) and ESC_WR_ENABLE register (0030h)).

Address: 4401 0021h

Bit	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	PROTECT
Value after reset	X	X	X	X	X	X	X	0

Table 5.16 WR_REG_PROTECT Register Contents

Bit Position	Bit Name	Function	R/W
b7 to b1	Reserved	When read, the value returned is undefined. When writing to these bits, write 0. Accessing from EtherCAT master: Read only	R
b0	PROTECT	Protection of registers against writing 0: Protection disabled 1: Protection enabled Accessing from EtherCAT master: R/W	R

5.4.13 ESC_WR_ENABLE — ESC Write Enable Register

This register is used to unlock the write protection temporarily while registers and memories are write protected by ESC write protection.

Address: 4401 0030h

Bit	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	ENABLE
Value after reset	X	X	X	X	X	X	X	0

Table 5.17 ESC_WR_ENABLE Register Contents

Bit Position	Bit Name	Function	R/W
b7 to b1	Reserved	When read, the value returned is undefined. When writing to these bits, write 0. Accessing from EtherCAT master: Read only	R
b0	ENABLE	When registers are currently being protected against writing by ESC write protection (bit 0 is 1 in the ESC write protection register, ESC_WR_PROTECT, at 0031h) and freely writing to registers of the given node is to be permitted, the operation to do so by writing to this register has to proceed in the same Ethernet frame and preceding the other desired writing to registers. Write protection will be reactivated once the frame period elapses (unless the value in the ESC write protection register is changed). Accessing from EtherCAT master: R/W	R

5.4.14 ESC_WR_PROTECT — ESC Write Protection Register

This register is used to protect registers against writing. Registers and memories in the area 4401 0000h to 4401 2FFFh including the process data RAM are write protected (except for the WR_REG_ENABLE register (0020h) and ESC_WR_ENABLE register (0030h)).

Address: 4401 0031h

Bit	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	PROTECT
Value after reset	X	X	X	X	X	X	X	0

Table 5.18 ESC_WR_PROTECT Register Contents

Bit Position	Bit Name	Function	R/W
b7 to b1	Reserved	When read, the value returned is undefined. When writing to these bits, write 0. Accessing from EtherCAT master: Read only	R
b0	PROTECT	Protection of registers and process memories against writing 0: Protection disabled 1: Protection enabled Accessing from EtherCAT master: R/W	R

5.4.15 ESC_RESET_ECAT — ESC Reset EtherCAT Register

This register is used to reset the EtherCAT slave controller from the EtherCAT (master) by software.

Address: 4401 0040h

Bit	b7	b6	b5	b4	b3	b2	b1	b0
	RESET_ECAT							
Value after reset	0	0	0	0	0	0	0	0

Table 5.19 ESC_RESET_ECAT Register Contents

Bit Position	Bit Name	Function	R/W
b7 to b0	RESET_ECAT	When written (ESC_RESET_ECAT_W): ● bit[7:0] RESET_ECAT A reset is enabled after writing 52h (“R”), 45h (“E”) and 53h (“S”) consecutively to this register. Accessing from EtherCAT master: R/W When read (ESC_RESET_ECAT_R): ● bit[7:2] Reserved When read, the value returned is undefined. When writing to these bits, write 0. Accessing from EtherCAT master: Read only ● bit[1:0] RESET_ECAT Progress of the reset procedure 01b: After writing 52h 10b: After writing 45h (if 52h was written before) 00b: Others Accessing from EtherCAT master: R/W	R

5.4.16 ESC_RESET_PDI — ESC Reset PDI Register

This register is used to reset the EtherCAT slave controller from the PDI (slave) by software.

Address: 4401 0041h

Bit	b7	b6	b5	b4	b3	b2	b1	b0
	RESET_PDI							
Value after reset	0	0	0	0	0	0	0	0

Table 5.20 ESC_RESET_PDI Register Contents

Bit Position	Bit Name	Function	R/W
b7 to b0	RESET_PDI	When written (ESC_RESET_PDI_W): ● bit[7:0] RESET_PDI A reset is enabled after writing 52h (“R”), 45h (“E”) and 53h (“S”) consecutively to this register. Accessing from EtherCAT master: Read only When read (ESC_RESET_PDI_R): ● bit[7:2] Reserved When read, the value returned is undefined. When writing to these bits, write 0. Accessing from CPU and EtherCAT master: Read only ● bit[1:0] RESET_PDI Progress of the reset procedure 01b: After writing 52h 10b: After writing 45h (if 52h was written before) 00b: Others Accessing from EtherCAT master: Read only	R/W

5.4.17 ESC_DL_CONTROL — ESC DL Control Register

This register is used to control loop in the EtherCAT slave controller and configure the RX FIFO size and station alias.

Changes to loop configurations are delayed until any current reception or transmission of a frame through the port is completed.

Reducing the size of the RX FIFO depends on all masters and slaves connected to the same network as the EtherCAT having very precise clock sources. An RX FIFO size of 7 (default) is sufficient if the precision of all clocks is 100 ppm or better and RX FIFO sizes of 0 to 3 are possible if the precision is 25 ppm or better.

Address: 4401 0100h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	STAALIAS	—	—	—	—	—	RXFIFO		
Value after reset	X	X	X	X	X	X	X	0	X	X	X	X	X	1	1	1

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	LP3		LP2		LP1		LP0		—	—	—	—	—	—	TEMPUSE	FWDRULE
Value after reset	1	1	0	0	0	0	0	0	X	X	X	X	X	X	0	1

Table 5.21 ESC_DL_CONTROL Register Contents (1/2)

Bit Position	Bit Name	Function	R/W
b31 to b25	Reserved	When read, the value returned is undefined. When writing to these bits, write 0. Accessing from EtherCAT master: Read only	R
b24	STAALIAS	Station alias 0: Ignore station alias 1: Alias can be used for all configured address command types (FPRD, FPWR, ...). Accessing from EtherCAT master: R/W	R
b23 to b19	Reserved	When read, the value returned is undefined. When writing to these bits, write 0. Accessing from EtherCAT master: Read only	R
b18 to b16	RXFIFO	Set the RX FIFO size. The transfer time can be reduced by reducing the FIFO size. 0 to 3: –40 ns 4 to 6: No change 7: Default Accessing from EtherCAT master: R/W	R
b15, b14	LP3	Loop port 3 configuration (port 3 is not available on this LSI.) 00b: Auto 01b: Auto close 10b: Open 11b: Closed Accessing from EtherCAT master: R/W	R
b13, b12	LP2	Loop port 2 configuration 00b: Auto 01b: Auto close 10b: Open 11b: Closed Accessing from EtherCAT master: R/W	R

Table 5.21 ESC_DL_CONTROL Register Contents (2/2)

Bit Position	Bit Name	Function	R/W
b11, b10	LP1	Loop port 1 configuration 00b: Auto 01b: Auto close 10b: Open 11b: Closed Accessing from EtherCAT master: R/W	R
b9, b8	LP0	Loop port 0 configuration 00b: Auto 01b: Auto close 10b: Open 11b: Closed Accessing from EtherCAT master: R/W	R
b7 to b2	Reserved	When read, the value returned is undefined. When writing to these bits, write 0. Accessing from EtherCAT master: Read only	R
b1	TEMPUSE	Temporary use of bits 15 to 8 settings 0: Permanent use 1: Use for about 1 second, then revert to previous settings Accessing from EtherCAT master: R/W	R
b0	FWDRULE	Forwarding rule 0: EtherCAT frames are processed. Non EtherCAT frames are forwarded without processing. 1: EtherCAT frames are processed. Non EtherCAT frames are destroyed. The source MAC address is changed for every frame (SOURCE_MAC[1] is set to 1 (locally administered address)) regardless of the forwarding rule. Accessing from EtherCAT master: R/W	R

5.4.18 PHYSICAL_RW_OFFSET — Physical Read/Write Offset Register

This register is used to set the offset between read address and write address in the R/W commands.

Address: 4401 0108h

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RWOFFSET															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 5.22 PHYSICAL_RW_OFFSET Register Contents

Bit Position	Bit Name	Function	R/W
b15 to b0	RWOFFSET	Offset of R/W commands (FPRW, APRW) between read address and write address. That is, in the case of reading, RD_ADR = ADR (the given address is read) In the case of writing, WR_ADR = ADR + R/W offset (writing is to the address obtained by adding the offset set in this register to the given address) Accessing from EtherCAT master: R/W	R

5.4.19 ESC_DL_STATUS — ESC DL Status Register

This register indicates the state of the EtherCAT slave controller.

Reading this register from the EtherCAT clears bit 2 of the EtherCAT event request register (ECAT_EVENT_REQ at 0210h).

Address: 4401 0110h

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	COMP3	LP3	COMP2	LP2	COMP1	LP1	COMP0	LP0	PHYP3	PHYP2	PHYP1	PHYP0	—	ENHLINKD	PDIWDST	PDIOPE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	X	1	0	0

Table 5.23 ESC_DL_STATUS Register Contents (1/2)

Bit Position	Bit Name	Function	R/W
b15	COMP3	Communication on port 3 (port 3 is not available on this LSI.) 0: No stable communication 1: Communication established Accessing from EtherCAT master: Read only (ack)	R
b14	LP3	Loop port 3 (port 3 is not available on this LSI.) 0: Open 1: Closed Accessing from EtherCAT master: Read only (ack)	R
b13	COMP2	Communication on port 2 0: No stable communication 1: Communication established Accessing from EtherCAT master: Read only (ack)	R
b12	LP2	Loop port 2 0: Open 1: Closed Accessing from EtherCAT master: Read only (ack)	R
b11	COMP1	Communication on port 1 0: No stable communication 1: Communication established Accessing from EtherCAT master: Read only (ack)	R
b10	LP1	Loop port 1 0: Open 1: Closed Accessing from EtherCAT master: Read only (ack)	R
b9	COMP0	Communication on port 0 0: No stable communication 1: Communication established Accessing from EtherCAT master: Read only (ack)	R
b8	LP0	Loop port 0 0: Open 1: Closed Accessing from EtherCAT master: Read only (ack)	R
b7	PHYP3	Physical link on port 3 (port 3 is not available on this LSI.) 0: No link 1: Link detected Accessing from EtherCAT master: Read only (ack)	R

Table 5.23 ESC_DL_STATUS Register Contents (2/2)

Bit Position	Bit Name	Function	R/W
b6	PHYP2	Physical link on port 2 0: No link 1: Link detected Accessing from EtherCAT master: Read only (ack)	R
b5	PHYP1	Physical link on port 1 0: No link 1: Link detected Accessing from EtherCAT master: Read only (ack)	R
b4	PHYP0	Physical link on port 0 0: No link 1: Link detected Accessing from EtherCAT master: Read only (ack)	R
b3	Reserved	When read, the value returned is undefined. Accessing from EtherCAT master: Read only (ack)	R
b2	ENHLINKD	Enhanced link detection 0: Deactivated for all ports 1: Activated for at least one port ***Note)*** This bit is set to the value of bit 9 at address 0000h in the EEPROM the first time the EEPROM is loaded after power is initially supplied or after a reset. Accessing from EtherCAT master: Read only (ack)	R
b1	PDIWDST	PDI watchdog timer status 0: Timeout of the watchdog timer 1: Watchdog timer reloaded Accessing from EtherCAT master: Read only (ack)	R
b0	PDIOPE	PDI operation/EEPROM load state 0: EEPROM not loaded, the PDI not operational (process data RAM is not accessible) 1: EEPROM loaded correctly, the PDI operational (process data RAM is accessible) Accessing from EtherCAT master: Read only (ack)	R

5.4.20 AL_CONTROL — AL Control Register

This register is used to change the state transition of the device state machine and to acknowledge error indication.

The PDI has to read the AL control register after the EtherCAT has written it. Otherwise the EtherCAT cannot write again to the AL control register. Reading the AL control register from the PDI clears bit 0 of the AL event request register (AL_EVENT_REQ at 0220h).

Address: 4401 0120h

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	DEVICE ID	ERRIN DACK	INISTATE			
Value after reset	X	X	X	X	X	X	X	X	X	X	0	0	0	0	0	1

Table 5.24 AL_CONTROL Register Contents

Bit Position	Bit Name	Function	R/W
b15 to b6	Reserved	When read, the value returned is undefined. Accessing from EtherCAT master: R (/W)	R
b5	DEVICEID	Device ID request 0: No request 1: Request Accessing from EtherCAT master: R (/W)	R (clear)
b4	ERRINDACK	Error indication acknowledge (response) 0: Error Indication in AL status register is not acknowledged. 1: Error Indication in AL status register is acknowledged. Accessing from EtherCAT master: R (/W)	R (clear)
b3 to b0	INISTATE	Change the state transition of the device state machine. 1: Initial state request 3: Bootstrap state request 2: Pre-operational state request 4: Safe-operational state request 8: Operational state request Accessing from EtherCAT master: R (/W)	R (clear)

5.4.21 AL_STATUS — AL Status Register

This register indicates the state of slave application.

Reading this register from the EtherCAT master clears bit 3 of the EtherCAT event request register (ECAT_EVENT_REQ at 0210h).

Address: 4401 0130h

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	DEVICE ID	ERR	ACTSTATE			
Value after reset	X	X	X	X	X	X	X	X	X	X	0	0	0	0	0	1

Table 5.25 AL_STATUS Register Contents

Bit Position	Bit Name	Function	R/W
b15 to b6	Reserved	When read, the value returned is undefined. When writing to these bits, write 0. Accessing from EtherCAT master: Read only (ack)	R/W
b5	DEVICEID	Status of loading device ID 0: Loading device ID failure 1: Loading device ID success Accessing from EtherCAT master: Read only (ack)	R/W
b4	ERR	Error indicator 0: The device is in the state as requested or flag was cleared by command. 1: The device has not entered the requested state or the state was changed as a result of local action. Accessing from EtherCAT master: Read only (ack)	R/W
b3 to b0	ACTSTATE	Actual state of the device state machine 1: Initial state 3: Request bootstrap state 2: Pre-operational state 4: Safe-operational state 8: Operational state Accessing from EtherCAT master: Read only (ack)	R/W

5.4.22 AL_STATUS_CODE — AL Status Code Register

This register indicates an error code from slave application.

Address: 4401 0134h

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	STATUSCODE															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 5.26 AL_STATUS_CODE Register Contents

Bit Position	Bit Name	Function	R/W
b15 to b0	STATUSCODE	AL status code Accessing from EtherCAT master: R	R/W

5.4.23 RUN_LED_OVERRIDE — RUN LED Override Register

This register is used to override control of the RUN LED pin.

Changing the value of the AL status register to an appropriate value will clear bit 4 (override enable).

Normally RUN LED is controlled by the AL status register (AL_STATUS at 0130h) automatically. It is not necessary to override RUN LED in order to indicate the state of a general state machine.

For example, this register can be used to run special lighting patterns that indicate the positions of specific slaves.

Address: 4401 0138h

Bit	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	OVERRIDEEN	LEDCODE			
Value after reset	X	X	X	0	0	0	0	0

Table 5.27 RUN_LED_OVERRIDE Register Contents

Bit Position	Bit Name	Function	R/W
b7 to b5	Reserved	When read, the value returned is undefined. When writing to these bits, write 0. Accessing from EtherCAT master: R/W	R
b4	OVERRIDEEN	Override enable 0: Override disabled 1: Override enabled Accessing from EtherCAT master: R/W	R/W
b3 to b0	LEDCODE	LED code 0h: Off 1h – Ch: Flash 1x – 12x Dh: Blinking Eh: Flickering Fh: On Accessing from EtherCAT master: R/W	R/W

5.4.24 ERR_LED_OVERRIDE — ERR LED Override Register

This register is used to override control of the error LED pin.

Bit 4 (override enable) will be cleared if a new error occurs.

The ESC automatically controls an error LED under the conditions below. Regarding other errors, the error LED should be controlled by application using this register.

- SII EEPROM load error
- PDI watchdog timeout

Address: 4401 0139h

Bit	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	OVERRIDEEN	LEDCODE			
Value after reset	X	X	X	0	0	0	0	0

Table 5.28 ERR_LED_OVERRIDE Register Contents

Bit Position	Bit Name	Function	R/W
b7 to b5	Reserved	When read, the value returned is undefined. When writing to these bits, write 0. Accessing from EtherCAT master: R/W	R
b4	OVERRIDEEN	Override enable 0: Override disabled 1: Override enabled Accessing from EtherCAT master: R/W	R/W
b3 to b0	LEDCODE	LED code 0h: Off 1h to Ch: Flash 1x to 12x Dh: Blinking Eh: Flickering Fh: On Accessing from EtherCAT master: R/W	R/W

5.4.25 PDI_CONTROL — PDI Control Register

This register indicates the type of PDI.

Address: 4401 0140h

Bit	b7	b6	b5	b4	b3	b2	b1	b0
	PDI							
Value after reset	1	0	0	0	0	0	0	0

Table 5.29 PDI_CONTROL Register Contents

Bit Position	Bit Name	Function	R/W
b7 to b0	PDI	Process data interface. In this LSI, the below value is indicated. 80h: On-chip bus Accessing from EtherCAT master: Read only	R

5.4.26 ESC_CONFIG — ESC Configuration Register

This register indicates configuration of the EtherCAT slave controller.

Address: 4401 0141h

Bit	b7	b6	b5	b4	b3	b2	b1	b0
	ENLP3	ENLP2	ENLP1	ENLP0	DCLATCH	DCSYNC	ENLALLP	DEVEMU
Value after reset	0	0	0	0	1	1	0	0

Table 5.30 ESC_CONFIG Register Contents

Bit Position	Bit Name	Function	R/W
b7	ENLP3	Sets enhanced link detection for port 3 (port 3 is not available on this LSI). 0: Disabled (if bit 9 of address 0 in the EEPROM = 0) 1: Enabled Accessing from EtherCAT master: Read only	R
b6	ENLP2	Sets enhanced link detection for port 2. 0: Disabled (if bit 9 of address 0 in the EEPROM = 0) 1: Enabled Accessing from EtherCAT master: Read only	R
b5	ENLP1	Sets enhanced link detection for port 1. 0: Disabled (if bit 9 of address 0 in the EEPROM = 0) 1: Enabled Accessing from EtherCAT master: Read only	R
b4	ENLP0	Sets enhanced link detection for port 0. 0: Disabled (if bit 9 of address 0 in the EEPROM = 0) 1: Enabled Accessing from EtherCAT master: Read only	R
b3	DCLATCH	Sets the latch input unit for distributed clocks (fixed to 1 in this LSI). 0: Disabled (power saving) 1: Enabled Accessing from EtherCAT master: Read only	R
b2	DCSYNC	Sets the SYNC output unit for distributed clocks (fixed to 1 in this LSI). 0: Disabled (power saving) 1: Enabled Accessing from EtherCAT master: Read only	R
b1	ENLALLP	Sets enhanced link detection for all ports. 0: Disabled (if bits 15 to 12 of address 0 in the EEPROM = 0) 1: Enabled at all ports Accessing from EtherCAT master: Read only	R
b0	DEVEMU	Device emulation (control of AL status) 0: The AL status register has to be set by the PDI. 1: The AL status register will be set to a value written to the AL control register. Accessing from EtherCAT master: Read only	R

5.4.27 PDI_CONFIG — PDI Configuration Register

This register indicates configuration of the PDI.

Address: 4401 0150h

Bit	b7	b6	b5	b4	b3	b2	b1	b0
	ONCHIPBUS			ONCHIPBUSCLK				
Value after reset	0	1	0	0	0	1	0	1

Table 5.31 PDI_CONFIG Register Contents

Bit Position	Bit Name	Function	R/W
b7 to b5	ONCHIPBUS	Indicate the type of on-chip bus. In this LSI, the value is always 010b. Accessing from EtherCAT master: Read only	R
b4 to b0	ONCHIPBUSCLK	Indicate the frequency of the on-chip bus clock. In this LSI, the value is always 5 (corresponding to 125 MHz). Accessing from EtherCAT master: Read only	R

5.4.28 SYNC_LATCH_CONFIG — SYNC/LATCH PDI Configuration Register

This register indicates the configuration of SYNC output and LATCH input.

Address: 4401 0151h

Bit	b7	b6	b5	b4	b3	b2	b1	b0
	SYNC1 MAP	SYNCL AT1	SYNC1OUT		SYNC0 MAP	SYNCL AT0	SYNC0OUT	
Value after reset	1	1	1	0	1	1	1	0

Table 5.32 SYNC_LATCH_CONFIG Register Contents

Bit Position	Bit Name	Function	R/W
b7	SYNC1MAP	Indicates enabling or disabling of mapping of the SYNC1 state to bit3 of the AL event request register (AL_EVENT_REQ at 0220h). This is always enabled in this LSI, so the value indicated is always 1(enabled). 0: Disabled 1: Enabled Accessing from EtherCAT master: Read only	R
b6	SYNCLAT1	Indicates the SYNC1/LATCH1 configuration. In this LSI, the value is always 1. Latch input is available, though the value indicates SYNC output. Use the IO Multiplexing function in order to switch SYNC output to LATCH input and vice versa. 0: LATCH1 input 1: SYNC1 output Accessing from EtherCAT master: Read only	R
b5, b4	SYNC1OUT	Indicate the SYNC1 output driver/polarity. In this LSI, the value is always 10b (push-pull active high). Accessing from EtherCAT master: Read only	R
b3	SYNC0MAP	Indicates enabling or disabling of mapping of the SYNC0 state to bit2 of the AL event request register (AL_EVENT_REQ at 0220h). This is always enabled in this LSI, so the value indicated is always 1(enabled). 0: Disabled 1: Enabled Accessing from EtherCAT master: Read only	R
b2	SYNCLAT0	Indicates the SYNC0/LATCH0 configuration. In this LSI, the value is always 1. Latch input is available, though the value indicates SYNC output. Use the IO Multiplexing function in order to switch SYNC output to LATCH input and vice versa. 0: LATCH0 input 1: SYNC0 output Accessing from EtherCAT master: Read only	R
b1, b0	SYNC0OUT	Indicate the SYNC0 output driver/polarity. In this LSI, the value is always 10b (push-pull active high). Accessing from EtherCAT master: Read only	R

5.4.29 EXT_PDI_CONFIG — Extended PDI Configuration Register

This register indicates configuration of the PDI.

Address: 4401 0152h

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DATABUSWID	
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	0	0

Table 5.33 EXT_PDI_CONFIG Register Contents

Bit Position	Bit Name	Function	R/W
b15 to b2	Reserved	When read, the value returned is undefined. Accessing from EtherCAT master: Read only	R
b1, b0	DATABUSWID	Indicate the data bus width of the PDI. In this LSI, the value is always 0 (4 bytes). 00b: 4 bytes 01b: 1 byte 10b: 2 bytes 11b: Reserved Accessing from EtherCAT master: Read only	R

5.4.30 ECAT_EVENT_MASK — EtherCAT Event Mask Register

The EtherCAT event request (EtherCAT interrupt) is used to transmit the slave event to the EtherCAT master. This register is used to set mask to each event of the EtherCAT event request register (ECAT_EVENT_REQ at 0210h). The logical AND of each effective bit in the EtherCAT event request register and the corresponding bit of this register is taken and the result produces the interrupt signal.

Address: 4401 0200h

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ECATEVMASK															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 5.34 ECAT_EVENT_MASK Register Contents

Bit Position	Bit Name	Function	R/W
b15 to b0	ECATEVMASK	0: The corresponding bit of the EtherCAT event request register (ECAT_EVENT_REQ at 0210h) is not mapped. 1: The corresponding bit of the EtherCAT event request register is mapped. Accessing from EtherCAT master: R/W	R

5.4.31 AL_EVENT_MASK — AL Event Mask Register

The AL event request (PDI interrupt) is used to transmit the ESC interrupt to the slave application. This register is used to set mask to each event of the AL event request register (AL_EVENT_REQ at 0220h). The logical AND of each effective bit in the AL event request register and the corresponding bit of this register is taken and the result produces the interrupt signal.

Address: 4401 0204h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	ALEVMASK															
Value after reset	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ALEVMASK															
Value after reset	1	1	1	1	1	1	1	1	0	0	0	0	1	1	1	1

Table 5.35 AL_EVENT_MASK Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	ALEVMASK	0: The corresponding bit of the AL event request register (AL_EVENT_REQ at 0220h) is not mapped. 1: The corresponding bit of the AL event request register is mapped. Accessing from EtherCAT master: Read only	R/W

5.4.32 ECAT_EVENT_REQ — EtherCAT Event Request Register

This register indicates the source of EtherCAT event requests (EtherCAT interrupts).

Address: 4401 0210h

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	SMSTA7	SMSTA6	SMSTA5	SMSTA4	SMSTA3	SMSTA2	SMSTA1	SMSTA0	ALSTA	DLSTA	—	DCLATCH
Value after reset	X	X	X	X	0	0	0	0	0	0	0	0	0	0	X	0

Table 5.36 ECAT_EVENT_REQ Register Contents (1/2)

Bit Position	Bit Name	Function	R/W
b15 to b12	Reserved	When read, the value returned is undefined. Accessing from EtherCAT master: Read only	R
b11	SMSTA7	Mirror value of SyncManager 7 status 0: No Sync channel 7 event 1: Sync channel 7 event pending Accessing from EtherCAT master: Read only	R
b10	SMSTA6	Mirror value of SyncManager 6 status 0: No Sync channel 6 event 1: Sync channel 6 event pending Accessing from EtherCAT master: Read only	R
b9	SMSTA5	Mirror value of SyncManager 5 status 0: No Sync channel 5 event 1: Sync channel 5 event pending Accessing from EtherCAT master: Read only	R
b8	SMSTA4	Mirror value of SyncManager 4 status 0: No Sync channel 4 event 1: Sync channel 4 event pending Accessing from EtherCAT master: Read only	R
b7	SMSTA3	Mirror value of SyncManager 3 status 0: No Sync channel 3 event 1: Sync channel 3 event pending Accessing from EtherCAT master: Read only	R
b6	SMSTA2	Mirror value of SyncManager 2 status 0: No Sync channel 2 event 1: Sync channel 2 event pending Accessing from EtherCAT master: Read only	R
b5	SMSTA1	Mirror value of SyncManager 1 status 0: No Sync channel 1 event 1: Sync channel 1 event pending Accessing from EtherCAT master: Read only	R
b4	SMSTA0	Mirror value of SyncManager 0 status 0: No Sync channel 0 event 1: Sync channel 0 event pending Accessing from EtherCAT master: Read only	R

Table 5.36 ECAT_EVENT_REQ Register Contents (2/2)

Bit Position	Bit Name	Function	R/W
b3	ALSTA	AL status event 0: No change in AL status 1: AL status change This bit is cleared by reading out the AL status register (AL_STATUS at 0130h or 0131h) from the EtherCAT. Accessing from EtherCAT master: Read only	R
b2	DLSTA	DL status event 0: No change in DL status 1: DL status change This bit is cleared by reading out the DL status register (ESC_DL_STATUS at 0110h or 0111h) from the EtherCAT. Accessing from EtherCAT master: Read only	R
b1	Reserved	When read, the value returned is undefined. Accessing from EtherCAT master: Read only	R
b0	DCLATCH	DC latch event 0: No change on DC latch Inputs 1: At least one change on DC latch Inputs This bit is cleared by reading DC latch event times from the EtherCAT master for EtherCAT controlled latch units, so the latch 0 and 1 status registers (DC_LATCH0_STAT and DC_LATCH1_STAT at 09AEh and 09AFh) indicate no event. Accessing from EtherCAT master: Read only	R

5.4.33 AL_EVENT_REQ — AL Event Request Register

This register indicates the source of AL event requests (PDI interrupts).

Address: 4401 0220h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SMINT7	SMINT6	SMINT5	SMINT4	SMINT3	SMINT2	SMINT1	SMINT0	—	WDPD	—	SYNCACT	DCSYNC1STA	DCSYNC0STA	DCLATCH	ALCTRL
Value after reset	0	0	0	0	0	0	0	0	X	0	X	0	0	0	0	0

Table 5.37 AL_EVENT_REQ Register Contents (1/2)

Bit Position	Bit Name	Function	R/W
b31 to b16	Reserved	When read, the value returned is undefined. Accessing from EtherCAT master: Read only	R
b15	SMINT7	SyncManager 7 interrupt (bit 0 or 1 of the SyncManager status 7 register (083Dh)) 0: No SyncManager 7 interrupt 1: SyncManager 7 interrupt pending Accessing from EtherCAT master: Read only	R
b14	SMINT6	SyncManager 6 interrupt (bit 0 or 1 of the SyncManager status 6 register (0835h)) 0: No SyncManager 6 interrupt 1: SyncManager 6 interrupt pending Accessing from EtherCAT master: Read only	R
b13	SMINT5	SyncManager 5 interrupt (bit 0 or 1 of the SyncManager status 5 register (082Dh)) 0: No SyncManager 5 interrupt 1: SyncManager 5 interrupt pending Accessing from EtherCAT master: Read only	R
b12	SMINT4	SyncManager 4 interrupt (bit 0 or 1 of the SyncManager status 4 register (0825h)) 0: No SyncManager 4 interrupt 1: SyncManager 4 interrupt pending Accessing from EtherCAT master: Read only	R
b11	SMINT3	SyncManager 3 interrupt (bit 0 or 1 of the SyncManager status 3 register (081Dh)) 0: No SyncManager 3 interrupt 1: SyncManager 3 interrupt pending Accessing from EtherCAT master: Read only	R
b10	SMINT2	SyncManager 2 interrupt (bit 0 or 1 of the SyncManager status 2 register (0815h)) 0: No SyncManager 2 interrupt 1: SyncManager 2 interrupt pending Accessing from EtherCAT master: Read only	R
b9	SMINT1	SyncManager 1 interrupt (bit 0 or 1 of the SyncManager status 1 register (080Dh)) 0: No SyncManager 1 interrupt 1: SyncManager 1 interrupt pending Accessing from EtherCAT master: Read only	R
b8	SMINT0	SyncManager 0 interrupt (bit 0 or 1 of the SyncManager status 0 register (0805h)) 0: No SyncManager 0 interrupt 1: SyncManager 0 interrupt pending Accessing from EtherCAT master: Read only	R

Table 5.37 AL_EVENT_REQ Register Contents (2/2)

Bit Position	Bit Name	Function	R/W
b7	Reserved	When read, the value returned is undefined. Accessing from EtherCAT master: Read only	R
b6	WDPD	Watchdog process data 0: Valid 1: Timeout This bit is cleared by reading the watchdog status process data register (WDS_DATA at 0440h) from the PDI. Accessing from EtherCAT master: Read only	R
b5	Reserved	When read, the value returned is undefined. Accessing from EtherCAT master: Read only	R
b4	SYNCACT	Change of the SyncManager activate [n] register (SM[n]_ACT at 0806h + 8h*n) 0: No change in any SyncManager 1: At least one SyncManager changed This bit is cleared by reading SyncManager activate [n] registers from the PDI. Accessing from EtherCAT master: Read only	R
b3	DCSYNC1STA	State of DC SYNC1 This bit is cleared by reading the SYNC1 status register (DC_SYNC1_STAT at 098Fh) from the PDI. Accessing from EtherCAT master: Read only	R
b2	DCSYNC0STA	State of DC SYNC0 This bit is cleared by reading the SYNC0 status register (DC_SYNC0_STAT at 098Eh) from the PDI. Accessing from EtherCAT master: Read only	R
b1	DCLATCH	DC latch event 0: No change on DC latch Inputs 1: At least one change on DC latch inputs This bit is cleared by reading DC latch event times from the PDI for PDI controlled latch units, so the latch 0 and 1 status registers (DC_LATCH0_STAT and DC_LATCH1_STAT at 09AEh and 09AFh) indicate no event. Accessing from EtherCAT master: Read only	R
b0	ALCTRL	AL control event 0: No change in the AL control register 1: The AL control register has been written. This bit is cleared by reading the AL control register (AL_CONTROL at 0120h or 0121h) from the PDI. Accessing from EtherCAT master: Read only	R

5.4.34 RX_ERR_COUNT[n] — Rx Error Counter [n] Register (n = 0..2)

This register counts RX frame errors.

(Port0 of ESC → PortA, Port1 of ESC → PortB, Port2 of ESC → PortC)

Address: 4401 0300h + 2h × n

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RXERRCNT								INVFRMCNT							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 5.38 RX_ERR_COUNT[n] Register Contents

Bit Position	Bit Name	Function	R/W
b15 to b8	RXERRCNT	Counter value of RX errors for port[n] Counting is stopped when FFh is reached. The number of RX errors of MII interface is counted. These bits are cleared if one of the RX error counters (RX_ERR_COUNT[n], FWD_RX_ERR_COUNT[n]) is written. Accessing from EtherCAT master: R/W (clear)	R
b7 to b0	INVFRMCNT	Counter value of invalid frames for port[n] Counting is stopped when FFh is reached. These bits are cleared if one of the RX error counters (RX_ERR_COUNT[n], FWD_RX_ERR_COUNT[n]) is written. Accessing from EtherCAT master: R/W (clear)	R

5.4.35 FWD_RX_ERR_COUNT[n] — Forwarded Rx Error Counter [n] Register (n = 0..2)

This register counts forwarded RX frame errors.

(Port0 of ESC → PortA, Port1 of ESC → PortB, Port2 of ESC → PortC)

Address: 4401 0308h + 1h × n

Bit	b7	b6	b5	b4	b3	b2	b1	b0
	FWDERRCNT							
Value after reset	0	0	0	0	0	0	0	0

Table 5.39 FWD_RX_ERR_COUNT[n] Register Contents

Bit Position	Bit Name	Function	R/W
b7 to b0	FWDERRCNT	Counter value of forwarded RX error frames for port[n] Counting is stopped when FFh is reached. These bits are cleared if one of the RX error counters (RX_ERR_COUNT[n], FWD_RX_ERR_COUNT[n]) is written. Accessing from EtherCAT master: R/W (clear)	R

5.4.36 ECAT_PROC_ERR_COUNT — EtherCAT Processing Unit Error Counter Register

This register counts frame errors passing the EtherCAT processing unit.

Address: 4401 030Ch

Bit	b7	b6	b5	b4	b3	b2	b1	b0
	EPUERRCNT							
Value after reset	0	0	0	0	0	0	0	0

Table 5.40 ECAT_PROC_ERR_COUNT Register Contents

Bit Position	Bit Name	Function	R/W
b7 to b0	EPUERRCNT	EtherCAT processing unit error counter value Counting is stopped when FFh is reached. This register counts errors of frames passing the processing unit. Writing to this register clears it. Accessing from EtherCAT master: R/W (clear)	R

5.4.37 PDI_ERR_COUNT — PDI Error Counter Register

This register counts PDI access errors.

Address: 4401 030Dh

Bit	b7	b6	b5	b4	b3	b2	b1	b0
	PDIERRCNT							
Value after reset	0	0	0	0	0	0	0	0

Table 5.41 PDI_ERR_COUNT Register Contents

Bit Position	Bit Name	Function	R/W
b7 to b0	PDIERRCNT	PDI error counter value Counting is stopped when FFh is reached. Counting starts when an interface error occurs due to access to the PDI. Writing to this register clears it. Accessing from EtherCAT master: R/W (clear)	R

5.4.38 LOST_LINK_COUNT[n] — Lost Link Counter [n] Register (n = 0..2)

This register counts lost links at port [n].

(Port0 → PortA, Port1 → PortB, Port2 → PortC)

Address: 4401 0310h + 1h × n

Bit	b7	b6	b5	b4	b3	b2	b1	b0
	LOSTLINKCNT							
Value after reset	0	0	0	0	0	0	0	0

Table 5.42 LOST_LINK_COUNT[n] Register Contents

Bit Position	Bit Name	Function	R/W
b7 to b0	LOSTLINKCNT	Lost link counter value for port [n]. Counting is stopped when FFh is reached. Counting starts only when port loop is Auto or Auto-Close. Only lost links at open ports are counted. Writing to one of the lost link counter registers clears it. Accessing from EtherCAT master: R/W (clear)	R

5.4.39 WD_DIVIDE — Watchdog Divider Register

This register is used to set the ratio for dividing 25 MHz to obtain the basic period for incrementing the watchdog timer.

Address: 4401 0400h

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	WDDIV															
Value after reset	0	0	0	0	1	0	0	1	1	1	0	0	0	0	1	0

Table 5.43 WD_DIVIDE Register Contents

Bit Position	Bit Name	Function	R/W
b15 to b0	WDDIV	Set the frequency divisor of the clock to drive counting by the watchdog timer in units of ticks at 25 MHz. The clock that drives counting by the watchdog timer is obtained by dividing 25 MHz by the value in this register plus two. The default value is 2498, which corresponds to a period of 100 µs. Accessing from EtherCAT master: R/W	R

5.4.40 WDT_PDI — Watchdog Time PDI Register

This register is used to set the time until the PDI watchdog timer overflows.

Address: 4401 0410h

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	WDTIMPDI															
Value after reset	0	0	0	0	0	0	1	1	1	1	1	0	1	0	0	0

Table 5.44 WDT_PDI Register Contents

Bit Position	Bit Name	Function	R/W
b15 to b0	WDTIMPDI	Set the time until the PDI watchdog timer overflows as a number of times the watchdog is incremented. With the default values for these bits and the setting of the watchdog divider, the time for a single incrementation is 100 µs, so the watchdog timer overflows when 100 µs x 1000 = 100 ms elapse. Setting these bits to 0 disables the watchdog timer. Access to the PDI restarts the watchdog timer. Accessing from EtherCAT master: R/W	R

5.4.41 WDT_DATA — Watchdog Time Process Data Register

This register is used to set the time until the process data watchdog timer overflows.

Address: 4401 0420h

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	WDTIMPD															
Value after reset	0	0	0	0	0	0	1	1	1	1	1	0	1	0	0	0

Table 5.45 WDT_DATA Register Contents

Bit Position	Bit Name	Function	R/W
b15 to b0	WDTIMPD	Set the time until the process data watchdog timer overflows as a number of times the watchdog is incremented. With the default values for these bits and the setting of the watchdog divider, the time for a single incrementation is 100 µs, so the watchdog timer overflows when 100 µs x 1000 = 100 ms elapse. There is one Watchdog for all SyncManagers. Setting these bits to 0 disables the watchdog timer. Access to the watchdog trigger enable bit of SyncManager restarts the watchdog timer. Accessing from EtherCAT master: R/W	R

5.4.42 WDS_DATA — Watchdog Status Process Data Register

This register indicates the state of the process data watchdog timer.

Address: 4401 0440h

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	WDSTAPD
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	0

Table 5.46 WDS_DATA Register Contents

Bit Position	Bit Name	Function	R/W
b15 to b1	Reserved	When read, the value returned is undefined. Accessing from EtherCAT master: Read only	R (ack)
b0	WDSTAPD	Indicates the state of the process data watchdog timer triggered by SyncManagers. 0: The timeout period of the process data watchdog timer elapses. 1: The process data watchdog timer is active or disabled. Reading this register clears bit 6 of the AL event request register (AL_EVENT_REQ at 0220h). Accessing from EtherCAT master: Read only	R (ack)

5.4.43 WDC_DATA — Watchdog Counter Process Data Register

This register indicates the timeout counter value of the process data watchdog timer.

Address: 4401 0442h

Bit	b7	b6	b5	b4	b3	b2	b1	b0
	WDCNTPD							
Value after reset	0	0	0	0	0	0	0	0

Table 5.47 WDC_DATA Register Contents

Bit Position	Bit Name	Function	R/W
b7 to b0	WDCNTPD	Counter value of the process data watchdog timer Counting stops when FFh is reached. Counting starts on a timeout of the process data watchdog timer. Writing to one of the watchdog counter registers (WDC_DATA, WDC_PDI at 0442h and 0443h) clears the counter. Accessing from EtherCAT master: R/W (clear)	R

5.4.44 WDC_PDI — Watchdog Counter PDI Register

This register indicates the timeout counter value of the PDI watchdog timer.

Address: 4401 0443h

Bit	b7	b6	b5	b4	b3	b2	b1	b0
	WDCNTPDI							
Value after reset	0	0	0	0	0	0	0	0

Table 5.48 WDC_PDI Register Contents

Bit Position	Bit Name	Function	R/W
b7 to b0	WDCNTPDI	Counter value of the PDI watchdog timer Counting stops when FFh is reached. Counting starts on a timeout of the PDI watchdog timer. Writing to one of the watchdog counter registers (WDC_DATA, WDC_PDI at 0442h and 0443h) clears the counter. Accessing from EtherCAT master: R/W (clear)	R

5.4.45 EEP_CONF — EEPROM Configuration Register

EtherCAT controls the SII EEPROM interface if EEPROM configuration register EEP_CONF.CTRLPDI (0500h) = 0 and EEPROM PDI Access State register EEP_STATE.PDIACCEES (0501h) = 0, otherwise PDI controls the EEPROM interface.

Address: 4401 0500h

Bit	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	FORCEECAT	CTRLPDI
Value after reset	X	X	X	X	X	X	0	0

Table 5.49 EEP_CONF Register Contents

Bit Position	Bit Name	Function	R/W
b7 to b2	Reserved	When read, the value returned is undefined. When writing to these bits, write 0. Accessing from EtherCAT master: Read only	R
b1	FORCEECAT	Forcibly changes the right of access to the EEPROM by the EtherCAT. 0: No change 1: Reset bit 0 of the EEPROM PDI access state register (EEP_STATE at 0501h) to 0. That is, prohibit access to the EEPROM by the PDI. Accessing from EtherCAT master: R/W	R
b0	CTRLPDI	Specifies whether EEPROM control is offered to the PDI. 0: The PDI has no EEPROM control. 1: The PDI has EEPROM control. Accessing from EtherCAT master: R/W	R

5.4.46 EEP_STATE — EEPROM PDI Access State Register

This register is used to set the right of access to the EEPROM by the PDI.

Address: 4401 0501h

Bit	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	PDIACCEES
Value after reset	X	X	X	X	X	X	X	0

Table 5.50 EEP_STATE Register Contents

Bit Position	Bit Name	Function	R/W
b7 to b1	Reserved	When read, the value returned is undefined. When writing to these bits, write 0. Accessing from EtherCAT master: Read only	R
b0	PDIACCEES	Sets the right of access to the EEPROM. 0: Prohibits the PDI from access to the EEPROM. 1: The PDI has access to the EEPROM. Write access from the PDI is only possible when bit 0 is 1 and bit 1 is 0 in the EEPROM configuration register (EEP_CONF at 0500h). Accessing from EtherCAT master: Read only	R(/W)

5.4.47 EEP_CONT_STAT — EEPROM Control/Status Register

This register is used to control access to the EEPROM and indicate the status.

Write access depends on the assignment of the EEPROM interface (EtherCAT/PDI). Write access is generally blocked if the EEPROM interface is busy (bit 15 = 1).

Address: 4401 0502h

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	BUSY	WRENERR	ACKCMDERR	LOADSTA	CKSUMERR	COMMAND			PROMSIZE	READBYTE	—	—	—	—	—	ECATWREN
Value after reset	0	0	0	0	0	0	0	0	0	0	X	X	X	X	X	0

Table 5.51 EEP_CONT_STAT Register Contents (1/2)

Bit Position	Bit Name	Function	R/W
b15	BUSY	Indicates a busy state of the EEPROM interface. 0: The EEPROM Interface is idle. 1: The EEPROM Interface is busy. Accessing from EtherCAT master: Read only	R
b14	WRENERR	Indicates error of write enable. Error bits are cleared by writing 000b (or any valid command) to command bits (bit 10 to 8). 0: No error 1: Write command without write enable Accessing from EtherCAT master: Read only	R
b13	ACKCMDERR	Indicates error of acknowledge/command: Error bits are cleared by writing 000b (or any valid command) to command bits (bit 10 to 8). 0: No error 1: Missing EEPROM acknowledge or invalid command Accessing from EtherCAT master: Read only	R
b12	LOADSTA	Indicates EEPROM loading status. 0: EEPROM has been loaded and device information has no problem 1: EEPROM has not been loaded and device information is not available (EEPROM loading in progress or finished with a failure). Accessing from EtherCAT master: Read only	R
b11	CKSUMERR	Indicates checksum error in the ESC configuration area. 0: No error in the checksum 1: Error in the checksum Accessing from EtherCAT master: Read only	R
b10 to b8	COMMAND	Command EtherCAT write enable (bit 0) is self-cleared at the SOF of the next frame. Command bits (bit 10 to 8) are also self-cleared after the command is executed (EEPROM busy ends). Writing 000b to command bits (bit 10 to 8) will also clear the error bits (bit 14 and 13). Command bits (bit 10 to 8) are ignored if the acknowledge/command error (bit 13) is 1. [Write] Initiates the commands below. [Read] Indicates the currently executed command. Commands: 000b: No command/EEPROM idle (clear error bits) 001b: Read 010b: Write 100b: Reload Others: Reserved/invalid commands (must not be issued) Accessing from EtherCAT master: R(/W)	R(/W)

Table 5.51 EEP_CONT_STAT Register Contents (2/2)

Bit Position	Bit Name	Function	R/W
b7	PROMSIZE	Indicates the selected EEPROM algorithm. 0: 1 address byte (1-Kbit to 16-Kbit EEPROMs) 1: 2 address bytes (32-Kbit to 4-Mbit EEPROMs) Accessing from EtherCAT master: Read only	R
b6	READBYTE	Indicates supported EEPROM read bytes. 0: 4 bytes 1: 8 bytes Accessing from EtherCAT master: Read only	R
b5 to b1	Reserved	When read, the value returned is undefined. Accessing from EtherCAT master: Read only	R
b0	ECATWREN	EtherCAT write enable EtherCAT write enable (bit 0) is self-cleared at the SOF of the next frame. Command bits (bit 10 to 8) are also self-cleared after the command is executed (EEPROM busy ends). Writing 000b to command bits (bit 10 to 8) will also clear the error bits (bit 14 and 13). Command bits (bit 10 to 8) are ignored if the acknowledge/command error (bit 13) is 1. 0: Write requests are disabled. 1: Write requests are enabled. This bit is always 1 if the PDI has EEPROM control. Accessing from EtherCAT master: R(/W)	R

5.4.48 EEP_ADR — EEPROM Address Register

This register is used to set the EEPROM address to be accessed.

Write access depends on the assignment of the EEPROM interface (EtherCAT/PDI). Write access is generally blocked if the EEPROM interface is busy (bit 15 is 1 in the EEPROM control/status register, EEP_CONT_STAT, at 0502h).

Address: 4401 0504h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	ADDRESS															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ADDRESS															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 5.52 EEP_ADR Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	ADDRESS	EEPROM address 0: First word (= 16 bits) 1: Second word : Actually used EEPROM address bits: [9:0]: EEPROM size of up to 16 Kbits [17:0]: EEPROM size of 32 Kbits to 4 Mbits Accessing from EtherCAT master: R(/W)	R(/W)

5.4.49 EEP_DATA — EEPROM Data Register

This register is used to set write data to the EEPROM or indicates read data from the EEPROM. It can be written in 1-word (2 bytes) units and read in 2-word (4 bytes) units.

Write access depends on the assignment of the EEPROM interface (EtherCAT/PDI). Write access is generally blocked if the EEPROM interface is busy (bit 15 is 1 in the EEPROM control/status register, EEP_CONT_STAT, at 0502h).

Address: 4401 0508h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	HIDATA															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	LODATA															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 5.53 EEP_DATA Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b16	HIDATA	Data read from the EEPROM (upper 2 bytes) Accessing from EtherCAT master: Read only	R
b15 to b0	LODATA	Data to be written to the EEPROM or data read from the EEPROM (lower 2 bytes) Accessing from EtherCAT master: R(/W)	R(/W)

5.4.50 MII_CONT_STAT — MII Management Control/Status Register

This register is used to control the MII management interface and to indicate the status.

Write access depends on the assignment of the management interface (EtherCAT/PDI). Write access is generally blocked if the management interface is busy (bit 15 = 1 in this register).

NOTE

Write enable bit (bit 0) is self-cleared at the SOF of the next frame. Command bits (bit 9 to 8) are also self-cleared after the command is executed (busy ends).
Writing 00b to command bits will also clear the error bits (bit 14 and 13). Command bits are cleared after the command is executed.

Address: 4401 0510h

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	BUSY	CMDERR	READERR	—	—	—	COMMAND		PHYOFFSET					MILINK	PDICTRL	WREN
Value after reset	0	0	0	X	X	X	0	0	0	0	0	0	0	0	1	0

Table 5.54 MII_CONT_STAT Register Contents (1/2)

Bit Position	Bit Name	Function	R/W
b15	BUSY	Indicates that the MII management interface is busy. 0: MII management interface is idle. 1: MII management interface is busy. Accessing from EtherCAT master: Read only	R
b14	CMDERR	Indicates whether a command error occurred. 0: Last command was successful 1: Invalid command or write command without write enable This bit is cleared by executing a valid command or writing 00b to command bits (bit 9 and 8). Accessing from EtherCAT master: Read only	R
b13	READERR	Indicates whether a read error occurred. 0: No read error 1: Read error occurred (PHY or register not available) This bit is cleared by writing to this register. Accessing from EtherCAT master: R(/W)	R(/W)
b12 to b10	Reserved	When read, the value returned is undefined. Accessing from EtherCAT master: Read only	R
b9, b8	COMMAND	Command [Write] Initiates the commands below. [Read] Indicates the currently executed command. Commands: 00b: No command/MI idle (clear error bits) 01b: Read 10b: Write Others: Reserved/invalid commands (must not be issued) Accessing from EtherCAT master: R(/W)	R(/W)
b7 to b3	PHYOFFSET	Indicate the PHY address offset. Accessing from EtherCAT master: Read only	R

Table 5.54 MII_CONT_STAT Register Contents (2/2)

Bit Position	Bit Name	Function	R/W
b2	MILINK	MI link detection 0: Not available 1: Available Accessing from EtherCAT master: Read only	R
b1	PDICTRL	Indicates whether the MII management interface can be controlled by the PDI. 0: Only EtherCAT control 1: PDI control possible The interface is controlled by the MII management EtherCAT access state register (MII_ECAT_ACS_STAT at 0516h) and the MII management PDI access state register (MII_PDI_ACS_STAT at 0517h). Accessing from EtherCAT master: Read only	R
b0	WREN	Write enable 0: Disabled 1: Enabled This bit is always 1 if the PDI controls the MII management interface. Accessing from EtherCAT master: R(/W)	R

5.4.51 PHY_ADR — PHY Address Register

This register is used to set the PHY address.

Write access depends on the assignment of the management interface (EtherCAT/PDI). Write access is generally blocked if the management interface is busy (bit 15 is 1 in the MII management control/status register, MII_CONT_STAT, at 0510h).

Address: 4401 0512h

Bit	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	PHYADDR				
Value after reset	X	X	X	0	0	0	0	0

Table 5.55 PHY_ADR Register Contents

Bit Position	Bit Name	Function	R/W
b7 to b5	Reserved	When read, the value returned is undefined. Accessing from EtherCAT master: Read only	R
b4 to b0	PHYADDR	PHY address Accessing from EtherCAT master: R(/W)	R(/W)

5.4.52 PHY_REG_ADR — PHY Register Address Register

This register is used to set the PHY register address.

Write access depends on the assignment of the management interface (EtherCAT/PDI). Write access is generally blocked if the management interface is busy (bit 15 is 1 in the MII management control/status register, MII_CONT_STAT, at 0510h).

Address: 4401 0513h

Bit	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	PHYREGADDR				
Value after reset	X	X	X	0	0	0	0	0

Table 5.56 PHY_REG_ADR Register Contents

Bit Position	Bit Name	Function	R/W
b7 to b5	Reserved	When read, the value returned is undefined. Accessing from EtherCAT master: Read only	R
b4 to b0	PHYREGADDR	Address of PHY register Accessing from EtherCAT master: R(/W)	R(/W)

5.4.53 PHY_DATA — PHY Data Register

This register is used to set data to write to PHY registers or to indicate data read from PHY registers.

Write access depends on the assignment of the management interface (EtherCAT/PDI). Write access is generally blocked if the management interface is busy (bit 15 is 1 in the MII management control/status register, MII_CONT_STAT, at 0510h).

Address: 4401 0514h

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PHYREGDATA															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 5.57 PHY_DATA Register Contents

Bit Position	Bit Name	Function	R/W
b15 to b0	PHYREGDATA	PHY register read/write data Accessing from EtherCAT master: R(/W)	R(/W)

5.4.54 MII_ECAT_ACS_STAT — MII Management EtherCAT Access State Register

This register is used to set the right of access to the MII management interface.

Write access is only possible when bit 0 is 1 in the MII management PDI access state register (MII_PDI_ACS_STAT at 0517h).

Address: 4401 0516h

Bit	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	ACSMII
Value after reset	X	X	X	X	X	X	X	0

Table 5.58 MII_ECAT_ACS_STAT Register Contents

Bit Position	Bit Name	Function	R/W
b7 to b1	Reserved	When read, the value returned is undefined. Accessing from EtherCAT master: Read only	R
b0	ACSMII	Right of access to the MII management interface 0: Enables access to the MII management interface by the PDI 1: Exclusive access to the MII management interface by the ECAT Accessing from EtherCAT master: R(/W)	R

5.4.55 MII_PDI_ACS_STAT — MII Management PDI Access State Register

This register is used to set the right of access to the MII management interface.

Address: 4401 0517h

Bit	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	FORPDI	ACSMII
Value after reset	X	X	X	X	X	X	0	0

Table 5.59 MII_PDI_ACS_STAT Register Contents

Bit Position	Bit Name	Function	R/W
b7 to b2	Reserved	When read, the value returned is undefined. When writing to these bits, write 0. Accessing from EtherCAT master: Read only	R
b1	FORPDI	Forced change of access by the PDI (forced change of bit0) 0: The value of bit 0 is not changed. 1: The value of bit 0 is reset to 0 (the right of access is changed to the EtherCAT) Accessing from EtherCAT master: R/W	R
b0	ACSMII	Right of access to the MII management interface 0: Access to the MII management interface by the PDI 1: Access to the MII management interface by the EtherCAT Accessing from EtherCAT master: Read only Write access to bit 0 from the PDI is only possible if the following two conditions are satisfied. ● Bit 0 is 0 in the MII management EtherCAT access state register (MII_ECAT_ACS_STAT at 0516h). ● Bit 1 is 0 in the MII management PDI access state register (MII_PDI_ACS_STAT at 0517h).	R(/W)

5.4.56 PHY_STATUS[n] — PHY Port Status [n] Register (n = 0..2)

This register indicates the PHY status for each port.

(Port0 of ESC → PortA, Port1 of ESC → PortB, Port2 of ESC → PortC)

Address: 4401 0518h + 1h × n

Bit	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	PHYCONFIG	LINKPARTERR	READERR	LINKSTAERR	LINKSTA	PHYLINKSTA
Value after reset	X	X	0	0	0	0	0	0

Table 5.60 PHY_STATUS[n] Register Contents

Bit Position	Bit Name	Function	R/W
b7, b6	Reserved	When read, the value returned is undefined. Accessing from EtherCAT master: Read only	R
b5	PHYCONFIG	PHY configuration update 0: No update 1: PHY configuration was updated. This bit is cleared by writing any value to at least one of the PHY port status [n] registers (PHY_STATUS[n]). Accessing from EtherCAT master: R(/W/clear)	R(/W/clear)
b4	LINKPARTERR	Link partner error 0: No error detected 1: Link partner error Accessing from EtherCAT master: Read only	R
b3	READERR	Read error 0: No read error occurred. 1: A read error has occurred. This bit is cleared by writing any value to at least one of the PHY port status [n] registers (PHY_STATUS[n]). Accessing from EtherCAT master: R (/W/clear)	R(/W/clear)
b2	LINKSTAERR	Link status error 0: No error 1: Link error or link inhibited Accessing from EtherCAT master: Read only	R
b1	LINKSTA	Link status (at 100 Mbits/s, full duplex, auto-negotiation) 0: No link under the above conditions 1: Link detected under the above conditions Accessing from EtherCAT master: Read only	R
b0	PHYLINKSTA	Physical link status (bit 2 of PHY register 1 (status register)) 0: The physical link is down. 1: The physical link is up. Accessing from EtherCAT master: Read only	R

5.4.57 FMMU[n]_L_START_ADR — FMMU Logical Start Address [n] Register (n = 0..7)

This register is used to set the logical start address within the EtherCAT address space for FMMU[n].

Address: 4401 0600h + 10h × n

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	LSTAADR															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	LSTAADR															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 5.61 FMMU[n]_L_START_ADR Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	LSTAADR	Set the start of the logical address within the EtherCAT address space. Accessing from EtherCAT master: R/W	R

5.4.58 FMMU[n]_LEN — FMMU Length [n] Register (n = 0..7)

This register is used to set the length for FMMU[n] area in byte units.

Address: 4401 0604h + 10h × n

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	FMMULEN															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 5.62 FMMU[n]_LEN Register Contents

Bit Position	Bit Name	Function	R/W
b15 to b0	FMMULEN	Set the area size in byte units. Address at the end of the logical address range set by the FMMU[n] minus the address at the start of the logical address range set by the FMMU[n] plus one. Accessing from EtherCAT master: R/W	R

5.4.59 FMMU[n]_L_START_BIT — FMMU Logical Start Bit [n] Register (n = 0..7)

This register is used to set the start bits of the logical start address for FMMU[n].

Address: 4401 0606h + 10h × n

Bit	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	LSTABIT		
Value after reset	X	X	X	X	X	0	0	0

Table 5.63 FMMU[n]_L_START_BIT Register Contents

Bit Position	Bit Name	Function	R/W
b7 to b3	Reserved	When read, the value returned is undefined. When writing to these bits, write 0. Accessing from EtherCAT master: Read only	R
b2 to b0	LSTABIT	Set the start bits of the logical start address for FMMU[n]. Accessing from EtherCAT master: R/W	R

5.4.60 FMMU[n]_L_STOP_BIT — FMMU Logical Stop bit [n] Register (n = 0..7)

This register is used to set the last bits of the logical end address for FMMU[n].

Address: 4401 0607h + 10h × n

Bit	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	LSTPBIT		
Value after reset	X	X	X	X	X	0	0	0

Table 5.64 FMMU[n]_L_STOP_BIT Register Contents

Bit Position	Bit Name	Function	R/W
b7 to b3	Reserved	When read, the value returned is undefined. When writing to these bits, write 0. Accessing from EtherCAT master: Read only	R
b2 to b0	LSTPBIT	Set the last bits of the logical end address for FMMU[n]. Accessing from EtherCAT master: R/W	R

5.4.61 FMMU[n]_P_START_ADR — FMMU Physical Start Address [n] Register (n = 0..7)

This register is used to set the physical start address of the ESC to which the logical start address will be mapped by the FMMU[n].

Address: 4401 0608h + 10h × n

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PHYSTAADR															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 5.65 FMMU[n]_P_START_ADR Register Contents

Bit Position	Bit Name	Function	R/W
b15 to b0	PHYSTAADR	Set the physical start address to which the logical start address will be mapped. The address is set as an offset from the base address (4401 0000h). Accessing from EtherCAT master: R/W	R

5.4.62 FMMU[n]_P_START_BIT — FMMU Physical Start Bit [n] Register (n = 0..7)

This register is used to set the start bits of the physical start address of the ESC to which the start bits of the logical start address will be mapped by the FMMU[n].

Address: 4401 060Ah + 10h × n

Bit	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	PHYSTABIT		
Value after reset	X	X	X	X	X	0	0	0

Table 5.66 FMMU[n]_P_START_BIT Register Contents

Bit Position	Bit Name	Function	R/W
b7 to b3	Reserved	When read, the value returned is undefined. When writing to these bits, write 0. Accessing from EtherCAT master: Read only	R
b2 to b0	PHYSTABIT	Set the start bits of the physical start address to which the start bits of the logical start address will be mapped. Accessing from EtherCAT master: R/W	R

5.4.63 FMMU[n]_TYPE — FMMU Type [n] Register (n = 0..7)

This register is used to set the type of FMMU[n] access.

Address: 4401 060Bh + 10h × n

Bit	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	WRITE	READ
Value after reset	X	X	X	X	X	X	0	0

Table 5.67 FMMU[n]_TYPE Register Contents

Bit Position	Bit Name	Function	R/W
b7 to b2	Reserved	When read, the value returned is undefined. When writing to these bits, write 0. Accessing from EtherCAT master: Read only	R
b1	WRITE	Sets the mapping for write access. 0: Disabled 1: Enabled Accessing from EtherCAT master: R/W	R
b0	READ	Sets the mapping for read access. 0: Disabled 1: Enabled Accessing from EtherCAT master: R/W	R

5.4.64 FMMU[n]_ACT — FMMU Activate [n] Register (n = 0..7)

This register is used to enable or disable FMMU[n].

Address: 4401 060Ch + 10h × n

Bit	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	ACTIVATE
Value after reset	X	X	X	X	X	X	0	0

Table 5.68 FMMU[n]_ACT Register Contents

Bit Position	Bit Name	Function	R/W
b7 to b1	Reserved	When read, the value returned is undefined. When writing to these bits, write 0. Accessing from EtherCAT master: Read only	R
b0	ACTIVATE	Enables or disables FMMU[n]. 0: Disabled 1: Enabled Accessing from EtherCAT master: R/W	R

5.4.65 SM[n]_P_START_ADR — SyncManager Physical Start Address [n] Register (n = 0..7)

This register is used to set the physical start address for the area assigned to SyncManager[n].

This register can only be written when SyncManager is disabled (bit 0 is 0 in the SyncManager activate [n] register (SM[n]_ACT at 0806h + 8h × n)).

Address: 4401 0800h + 8h × n

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SMSTAADDR															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 5.69 SM[n]_P_START_ADR Register Contents

Bit Position	Bit Name	Function	R/W
b15 to b0	SMSTAADDR	Specify the physical start address for the area assigned to SyncManager. Accessing from EtherCAT master: R(/W)	R

5.4.66 SM[n]_LEN — SyncManager Length [n] Register (n = 0..7)

This register is used to set the size of the area assigned to SyncManager[n].

This register can only be written when SyncManager is disabled (bit 0 is 0 in the SyncManager activate [n] register (SM[n]_ACT at 0806h + 8h × n)).

Address: 4401 0802h + 8h × n

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SMLEN															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 5.70 SM[n]_LEN Register Contents

Bit Position	Bit Name	Function	R/W
b15 to b0	SMLEN	Set the number of bytes assigned to SyncManager. Set a value greater than 1. Otherwise, SyncManager is not activated. Accessing from EtherCAT master: R(/W)	R

5.4.67 SM[n]_CONTROL — SyncManager Control [n] Register (n = 0..7)

This register is used to set operation of SyncManager[n].

This register can only be written when SyncManager is disabled (bit 0 is 0 in the SyncManager activate [n] register (SM[n]_ACT at 0806h + 8h × n)).

Address: 4401 0804h + 8h × n

Bit	b7	b6	b5	b4	b3	b2	b1	b0
	—	WDTR GEN	IRQPDI	IRQEC AT	DIR		OPEMODE	
Value after reset	X	0	0	0	0	0	0	0

Table 5.71 SM[n]_CONTROL Register Contents

Bit Position	Bit Name	Function	R/W
b7	Reserved	When read, the value returned is undefined. Accessing from EtherCAT master: Read only	R
b6	WDTRGEN	Enables or disables watchdog trigger. 0: Disabled 1: Enabled Accessing from EtherCAT master: R(/W)	R
b5	IRQPDI	Enables or disables interrupts (PDI interrupts) by the AL event request register (AL_EVENT_REQ at 0220h). 0: Disabled 1: Enabled Accessing from EtherCAT master: R(/W)	R
b4	IRQECAT	Enables or disables interrupts (ECAT interrupts) by the EtherCAT event request register (ECAT_EVENT_REQ at 0210h). 0: Disabled 1: Enabled Accessing from EtherCAT master: R(/W)	R
b3 to b2	DIR	Transfer direction 00b: Read (EtherCAT master: read access; PDI: write access) 01b: Write (EtherCAT master: write access; PDI: read access) Others: Reserved Accessing from EtherCAT master: R(/W)	R
b1 to b0	OPEMODE	Operating mode 00b: Buffer mode (3 buffer mode) 10b: Mailbox mode (single buffer mode) Others: Reserved Accessing from EtherCAT master: R(/W)	R

5.4.68 SM[n]_STATUS — SyncManager Status [n] Register (n = 0..7)

This register indicates the state of SyncManager[n].

Address: 4401 0805h + 8h × n

Bit	b7	b6	b5	b4	b3	b2	b1	b0
	WRBUF	RDBUF	BUFFERED		MAILBOX	—	INTRD	INTWR
Value after reset	0	0	1	1	0	X	0	0

Table 5.72 SM[n]_STATUS Register Contents

Bit Position	Bit Name	Function	R/W
b7	WRBUF	Indicates that the buffer is being written. Accessing from EtherCAT master: Read only	R
b6	RDBUF	Indicates that the buffer is being read. Accessing from EtherCAT master: Read only	R
b5, b4	BUFFERED	Indicates the buffer status in buffer mode (last written buffer). 00b: 1st buffer 01b: 2nd buffer 10b: 3rd buffer 11b: No buffer written This bit is not used in mailbox mode. Accessing from EtherCAT master: Read only	R
b3	MAILBOX	Indicates the mailbox status in mailbox mode. 0: Mailbox empty 1: Mailbox full This bit is not used in buffer mode. Accessing from EtherCAT master: Read only	R
b2	Reserved	When read, the value returned is undefined. Accessing from EtherCAT master: Read only	R
b1	INTRD	Indicates read complete interrupt. 1: Indicates that the first byte of the buffer was written (interrupt cleared) 0: Indicates that the buffer was successfully read. Accessing from EtherCAT master: Read only	R
b0	INTWR	Indicates write complete interrupt. 1: Indicates that the first byte of the buffer was read (interrupt cleared) 0: Indicates that the buffer was successfully written. Accessing from EtherCAT master: Read only	R

5.4.69 SM[n]_ACT — SyncManager Activate [n] Register (n = 0..7)

This register is used to set operation of SyncManager[n].

Reading this register from the PDI in all SyncManagers which have changed activation clears bit 4 of the AL event request register (AL_EVENT_REQ at 0220h).

Address: 4401 0806h + 8h × n

Bit	b7	b6	b5	b4	b3	b2	b1	b0
	LATCH PDI	LATCH ECAT	—	—	—	—	REPEA TREQ	SMEN
Value after reset	0	0	X	X	X	X	0	0

Table 5.73 SM[n]_ACT Register Contents

Bit Position	Bit Name	Function	R/W
b7	LATCHPDI	PDI latch event 0: No 1: Generates latch events if the PDI switches the buffers or accesses the buffer start address. Accessing from EtherCAT master: R/W	R (ack)
b6	LATCHECAT	ECAT latch event 0: No 1: Generates latch events if the EtherCAT master switches the buffers. Accessing from EtherCAT master: R/W	R (ack)
b5 to b2	Reserved	Reserved/ When read, the value returned is undefined. When writing to these bits, write 0. Accessing from EtherCAT master: Read only	R (ack)
b1	REPEATREQ	Repeat request Toggling of the repeat request signal means that retrying the mailbox is required (primarily used in conjunction with reading of the ECAT mailbox). Accessing from EtherCAT master: R/W	R (ack)
b0	SMEN	Enables or disables SyncManager. 0: Disabled. Memory is accessed without SyncManager 1: Enabled. SyncManager is active and controls Memory area set in configuration Accessing from EtherCAT master: R/W	R (ack)

5.4.70 SM[n]_PDI_CONT — SyncManager PDI Control [n] Register (n = 0..7)

This register is used to set operation of SyncManager[n] from the PDI.

Address: 4401 0807h + 8h × n

Bit	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	REPEATACK	DEACTIVE
Value after reset	X	X	X	X	X	X	0	0

Table 5.74 SM[n]_PDI_CONT Register Contents

Bit Position	Bit Name	Function	R/W
b7 to b2	Reserved	When read, the value returned is undefined. When writing to these bits, write 0. Accessing from EtherCAT master: Read only	R
b1	REPEATACK	Repeat Acknowledge If this bit is set to the same value as set by bit 1 (repeat request) of the SyncManager activate [n] register (SM[n]_ACT), the PDI acknowledges the execution of a previous set repeat request. Accessing from EtherCAT master: Read only	R/W
b0	DEACTIVE	Deactivates SyncManager. [Read] 0: Normal operation. SyncManager is activated. 1: SyncManager is deactivated and reset. SyncManager locks access to memory area. [Write] 0: Activates SyncManager. 1: Deactivates SyncManager. ***Note)*** Writing 1 is delayed until the end of a frame which is currently processed Accessing from EtherCAT master: Read only	R/W

5.4.71 DC_RCV_TIME_PORT0 — Receive Times Port0 Register

Writing to this register latches the received time of frames at all ports and reading this register indicates the received time of a frame latched at port 0. (port A)

Address: 4401 0900h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	RCVTIME0															
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RCVTIME0															
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Table 5.75 DC_RCV_TIME_PORT0 Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	RCVTIME0	[Write] ● A write access to this register with the BWR, APWR (any address) or FPWR (configured address) command latches the local time when each port starts to receive a frame (first start bit of preamble). [Read] ● Indicate the local time at the beginning of reception of the last frame containing a write access to this register. ***Note*)** The time stamps cannot be read in the same frame in which this register was written. Accessing from EtherCAT master: R/W	R

5.4.72 DC_RCV_TIME_PORT1 — Receive Times Port1 Register

This register indicates the received time of the frame latched at port 1. (port B)

Address: 4401 0904h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	RCVTIME1															
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RCVTIME1															
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Table 5.76 DC_RCV_TIME_PORT1 Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	RCVTIME1	Indicate the local time when port 1 starts to receive a frame (first start bit of preamble) containing the BWR, APWR or FPWR command to the receive time port 0 register (DC_RCV_TIME_PORT0 at 0900h). Accessing from EtherCAT master: Read only	R

5.4.73 DC_RCV_TIME_PORT2 — Receive Times Port2 Register

This register indicates the received time of the frame latched at port 2. (port C)

Address: 4401 0908h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	RCVTIME2															
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RCVTIME2															
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Table 5.77 DC_RCV_TIME_PORT2 Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	RCVTIME2	Indicate the local time when port 2 starts to receive a frame (first start bit of preamble) containing the BWR, APWR or FPWR command to the receive time port 0 register (DC_RCV_TIME_PORT0 at 0900h). Accessing from EtherCAT master: Read only	R

5.4.74 DC_SYS_TIME — System Time Register

This register indicates the local copy of the system time.

Address: 4401 0910h

Bit	b63	b62	b61	b60	b59	b58	b57	b56	b55	b54	b53	b52	b51	b50	b49	b48
	SYSTIME															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b47	b46	b45	b44	b43	b42	b41	b40	b39	b38	b37	b36	b35	b34	b33	b32
	SYSTIME															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	SYSTIME															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SYSTIME															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 5.78 DC_SYS_TIME Register Contents

Bit Position	Bit Name	Function	R/W
b63 to b0	SYSTIME	Access from the ECAT [Read] ● Indicate the local copy of the system time when the frame passed the reference clock (i.e., including a system time delay). The time latched at the start of frame (SOF) is indicated. [Write] ● A written value is compared with the local copy of the system time. The result is input to the time control loop unit. Accessing from EtherCAT master: R/W Access from the PDI [Read] ● Indicate the local copy of the system time. The time latched when the first byte of this register was read is indicated. Accessing from EtherCAT master: Read only	R

5.4.75 DC_RCV_TIME_UNIT — Receive Time EtherCAT Processing Unit Register

This register indicates the received time of a frame latched at EtherCAT processing unit.

Address: 4401 0918h

Bit	b63	b62	b61	b60	b59	b58	b57	b56	b55	b54	b53	b52	b51	b50	b49	b48
	RCVTIMEEPU															
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b47	b46	b45	b44	b43	b42	b41	b40	b39	b38	b37	b36	b35	b34	b33	b32
	RCVTIMEEPU															
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	RCVTIMEEPU															
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RCVTIMEEPU															
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Table 5.79 DC_RCV_TIME_UNIT Register Contents

Bit Position	Bit Name	Function	R/W
b63 to b0	RCVTIMEEPU	This register indicates the local time at the beginning of reception by the EtherCAT processing unit of a frame (i.e. the first start bit of the preamble), including write access to the receive time port 0 register (DC_RCV_TIME_PORT0 at 0900h). Accessing from EtherCAT master: Read only	R

5.4.76 DC_SYS_TIME_OFFSET — System Time Offset Register

This register is used to indicate a difference (offset) between the local time and system time.

Address: 4401 0920h

Bit	b63	b62	b61	b60	b59	b58	b57	b56	b55	b54	b53	b52	b51	b50	b49	b48
	SYSTIMOFST															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b47	b46	b45	b44	b43	b42	b41	b40	b39	b38	b37	b36	b35	b34	b33	b32
	SYSTIMOFST															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	SYSTIMOFST															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SYSTIMOFST															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 5.80 DC_SYS_TIME_OFFSET Register Contents

Bit Position	Bit Name	Function	R/W
b63 to b0	SYSTIMOFST	Indicate a difference between the local time and system time. This offset is added to the local time to obtain the local system time. Accessing from EtherCAT master: R/W	R

5.4.77 DC_SYS_TIME_DELAY — System Time Delay Register

This register indicates a propagation delay between the reference clock and slave (ESC).

Address: 4401 0928h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	SYSTIMDLY															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SYSTIMDLY															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 5.81 DC_SYS_TIME_DELAY Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	SYSTIMDLY	Indicate a delay between the reference clock and the ESC. Accessing from EtherCAT master: R/W	R

5.4.78 DC_SYS_TIME_DIFF — System Time Difference Register

This register indicates a mean difference between the local copy of the system time and received system time.

Address: 4401 092Ch

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	LOCAL COPY	DIFF														
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DIFF															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 5.82 DC_SYS_TIME_DIFF Register Contents

Bit Position	Bit Name	Function	R/W
b31	LOCALCOPY	Indicates whether the local copy of the system time is greater than or equal to, or is less than, the latest received copy of the system time. 0: Local copy of the system time greater than or equal to the received system time 1: Local copy of the system time less than the received system time Accessing from EtherCAT master: Read only	R
b30 to b0	DIFF	Indicates a mean difference between the local copy of the system time and received system time. Accessing from EtherCAT master: Read only	R

5.4.79 DC_SPEED_COUNT_START — Speed Counter Start Register

This register is used to set the bandwidth for drift correction of the local copy of the system time.

Address: 4401 0930h

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	SPDCNTSTRT														
Value after reset	X	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0

Table 5.83 DC_SPEED_COUNT_START Register Contents

Bit Position	Bit Name	Function	R/W
b15	Reserved	When read, the value returned is undefined. When writing to this bit, write 0. Accessing from EtherCAT master: Read only	R
b14 to b0	SPDCNTSTRT	Indicate the bandwidth for adjustment of the local copy of the system time (larger values → smaller bandwidth and smoother adjustment). A write access resets the system time difference register (DC_SYS_TIME_DIFF at 092Ch) and the Speed Counter Diff register (DC_SPEED_COUNT_DIFF at 0932h). Valid range: 0080h to 3FFFh Accessing from EtherCAT master: R/W	R

5.4.80 DC_SPEED_COUNT_DIFF — Speed Counter Diff Register

This register indicates the deviation between the local clock period and reference clock's clock period.

Address: 4401 0932h

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SPDCNTDIFF															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 5.84 DC_SPEED_COUNT_DIFF Register Contents

Bit Position	Bit Name	Function	R/W
b15 to b0	SPDCNTDIFF	Indicate the deviation between the local clock period and reference clock's clock period (represented by two's complements). Range: ± (speed counter start value – 7Fh) Accessing from EtherCAT master: Read only	R

5.4.81 DC_SYS_TIME_DIFF_FIL_DEPTH — System Time Difference Filter Depth Register

This register is used to set the filter depth for averaging the received system time deviation.

Address: 4401 0934h

Bit	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	SYSTIMDEP			
Value after reset	X	X	X	X	0	1	0	0

Table 5.85 DC_SYS_TIME_DIFF_FIL_DEPTH Register Contents

Bit Position	Bit Name	Function	R/W
b7 to b4	Reserved	When read, the value returned is undefined. When writing to these bits, write 0. Accessing from EtherCAT master: Read only	R
b3 to b0	SYSTIMDEP	Set the filter depth for averaging the received system time deviation. A write access resets the system time difference register (DC_SYS_TIME_DIFF at 092Ch). Accessing from EtherCAT master: R/W	R

5.4.82 DC_SPEED_COUNT_FIL_DEPTH — Speed Counter Filter Depth Register

This register is used to set the filter depth for averaging the clock period deviation.

Address: 4401 0935h

Bit	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	CLKPERDEP			
Value after reset	X	X	X	X	1	1	0	0

Table 5.86 DC_SPEED_COUNT_FIL_DEPTH Register Contents

Bit Position	Bit Name	Function	R/W
b7 to b4	Reserved	When read, the value returned is undefined. When writing to these bits, write 0. Accessing from EtherCAT master: R/W	R
b3 to b0	CLKPERDEP	Set the filter depth for averaging the clock period deviation. A write access resets the internal speed counter filter. Accessing from EtherCAT master: R/W	R

5.4.83 DC_CYC_CONT — Cyclic Unit Control Register

This register sets whether to control SYNC and latch units by the ECAT or PDI.

Address: 4401 0980h

Bit	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	LATCH 1	LATCH 0	—	—	—	SYNCO UT
Value after reset	X	X	0	0	X	X	X	0

Table 5.87 DC_CYC_CONT Register Contents

Bit Position	Bit Name	Function	R/W
b7, b6	Reserved	When read, the value returned is undefined. When writing to these bits, write 0. Accessing from EtherCAT master: Read only	R
b5	LATCH1	Sets control of latch input unit 1. 0: EtherCAT control 1: PDI control ***Note*)** Latch interrupt is routed to the ECAT or PDI in accord with this setting. Accessing from EtherCAT master: R/W	R
b4	LATCH0	Sets control of latch input unit 0. 0: EtherCAT control 1: PDI control ***Note*)** Latch interrupt is routed to the ECAT or PDI in accord with this setting. Accessing from EtherCAT master: R/W	R
b3 to b1	Reserved	When read, the value returned is undefined. When writing to these bits, write 0. Accessing from EtherCAT master: Read only	R
b0	SYNCOUT	Sets control of the SYNC output unit. 0: EtherCAT control 1: PDI control Accessing from EtherCAT master: R/W	R

5.4.84 DC_ACT — Activation Register

This register is used to activate the Sync output unit.

Writing to this register depends on the setting of bit 0 of the cyclic unit control register (DC_CYC_CONT at 0980h).

Address: 4401 0981h

Bit	b7	b6	b5	b4	b3	b2	b1	b0
	DBGPULSE	NEARFUTURE	STARTTIME	EXTSTARTTIME	AUTOACT	SYNC1	SYNC0	SYNCACT
Value after reset	0	0	0	0	0	0	0	0

Table 5.88 DC_ACT Register Contents

Bit Position	Bit Name	Function	R/W
b7	DBGPULSE	Sets Sync signal debug pulse. 0: Deactivated 1: Immediately generates a single debug ping on the SYNC0 and SYNC1 pins in accord with the setting of bits 2 and 1 of this register. This bit is self-cleared and always read as 0. Accessing from EtherCAT master: R(/W)	R(/W)
b6	NEARFUTURE	Sets the range to be considered the near future. 0: Up to 2^{63} ns from now (1/2 of the DC width) 1: Up to 2^{31} ns from now (about 2.1 s) Accessing from EtherCAT master: R(/W)	R(/W)
b5	STARTTIME	Selects whether checking the plausibility of the start time and response to implausible start times is to proceed. 0: Disabled. Sync signal is generated if the start time is reached. 1: Sync signal is generated immediately if the start time is outside the range of the near future. Accessing from EtherCAT master: R(/W)	R(/W)
b4	EXTSTARTTIME	Extends start time cyclic operation. 0: No extension 1: Extends the start time written with 32 bits to 64 bits Accessing from EtherCAT master: R(/W)	R(/W)
b3	AUTOACT	Sets whether to activate the Sync output unit automatically by writing to the start time cyclic operation register (DC_CYC_START_TIME at 0990h): 0: Deactivated 1: Activated. Bit 0 is automatically set to 1 in this register after the start time is written. Accessing from EtherCAT master: R(/W)	R(/W)
b2	SYNC1	Sets SYNC1 output. 0: Deactivated 1: SYNC1 pulse output is generated. Accessing from EtherCAT master: R(/W)	R(/W)
b1	SYNC0	Sets SYNC0 output. 0: Deactivated 1: SYNC0 pulse output is generated. Accessing from EtherCAT master: R(/W)	R(/W)
b0	SYNCACT	Activates the Sync output unit. 0: Deactivated 1: Activated ***Note)*** Write 1 after the start time was written. Accessing from EtherCAT master: R(/W)	R(/W)

5.4.85 DC_PULSE_LEN — Pulse Length of SyncSignals Register

This register indicates the pulse length of SYNC signals.

Address: 4401 0982h

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PULSELEN															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 5.89 DC_PULSE_LEN Register Contents

Bit Position	Bit Name	Function	R/W
b15 to b0	PULSELEN	Indicate the pulse length of SYNC signals (in units of 10 ns) 0: Acknowledge mode. In this mode, SYNC signal is cleared by reading the SYNC0 or SYNC1 status register (DC_SYNC0/1_STAT at 098Eh, 098Fh). Accessing from EtherCAT master: Read only	R

5.4.86 DC_ACT_STAT — Activation Status Register

This register indicates the activation status of SYNC output signals.

Address: 4401 0984h

Bit	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	START TIME	SYNC1 ACT	SYNC0 ACT
Value after reset	X	X	X	X	X	0	0	0

Table 5.90 DC_ACT_STAT Register Contents

Bit Position	Bit Name	Function	R/W
b7 to b3	Reserved	When read, the value returned is undefined. Accessing from EtherCAT master: Read only	R
b2	STARTTIME	Indicates the plausibility check result of the start time cyclic operation register (DC_CYC_START_TIME at 0990h) while the Sync output unit is activated. 0: The start time was within the near future. 1: The start time was out of the near future. Accessing from EtherCAT master: Read only	R
b1	SYNC1ACT	Indicates the activation state of SYNC1. 0: First SYNC1 pulse is not pending. 1: First SYNC1 pulse is pending. Accessing from EtherCAT master: Read only	R
b0	SYNC0ACT	Indicates the activation state of SYNC0. 0: First SYNC0 pulse is not pending. 1: First SYNC0 pulse is pending. Accessing from EtherCAT master: Read only	R

5.4.87 DC_SYNC0_STAT — SYNC0 Status Register

This register indicates the state of SYNC0 output. It is only used in acknowledge mode.

Address: 4401 098Eh

Bit	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	SYNC0 STA
Value after reset	X	X	X	X	X	X	X	0

Table 5.91 DC_SYNC0_STAT Register Contents

Bit Position	Bit Name	Function	R/W
b7 to b1	Reserved	When read, the value returned is undefined. Accessing from EtherCAT master: Read only	R
b0	SYNC0STA	Indicates the SYNC0 state for acknowledge mode. SYNC0 in acknowledge mode is cleared by reading this register from the PDI. This bit is only used in acknowledge mode. Accessing from EtherCAT master: Read only	R (ack)

5.4.88 DC_SYNC1_STAT — SYNC1 Status Register

This register indicates the state of SYNC1 output. It is only used in acknowledge mode.

Address: 4401 098Fh

Bit	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	SYNC1 STA
Value after reset	X	X	X	X	X	X	X	0

Table 5.92 DC_SYNC1_STAT Register Contents

Bit Position	Bit Name	Function	R/W
b7 to b1	Reserved	When read, the value returned is undefined. Accessing from EtherCAT master: Read only	R
b0	SYNC1STA	Indicates the SYNC1 state for acknowledge mode. SYNC1 in acknowledge mode is cleared by reading this register from the PDI. This bit is only used in acknowledge mode. Accessing from EtherCAT master: Read only	R (ack)

5.4.89 DC_CYC_START_TIME — Start Time Cyclic Operation/Next SYNC0 Pulse Register

Writing to this register sets the start time of cyclic operation. Reading this register indicates the system time of the next SYNC0 pulse.

Writing to this register depends on the setting of bit 0 of the cyclic unit control register (DC_CYC_CONT at 0980h). Only writable when bit 0 is 0 in the SYNC activation register (DC_ACT at 0981h).

When auto activation is enabled, upper 32 bits are automatically extended if only lower 32 bits are written within one frame.

Address: 4401 0990h

Bit	b63	b62	b61	b60	b59	b58	b57	b56	b55	b54	b53	b52	b51	b50	b49	b48
	STATIM															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b47	b46	b45	b44	b43	b42	b41	b40	b39	b38	b37	b36	b35	b34	b33	b32
	STATIM															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	STATIM															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	STATIM															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 5.93 DC_CYC_START_TIME Register Contents

Bit Position	Bit Name	Function	R/W
b63 to b0	STATIM	[Write] ● Set the start time (in the system time) of cyclic operation in ns units. [Read] ● Indicate the system time of the next SYNC0 pulse in ns units. Accessing from EtherCAT master: R(/W)	R(/W)

5.4.90 DC_NEXT_SYNC1_PULSE — Next SYNC1 Pulse Register

This register indicates the system time of the next SYNC1 pulse.

Address: 4401 0998h

Bit	b63	b62	b61	b60	b59	b58	b57	b56	b55	b54	b53	b52	b51	b50	b49	b48
	SYNC1PULSE															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b47	b46	b45	b44	b43	b42	b41	b40	b39	b38	b37	b36	b35	b34	b33	b32
	SYNC1PULSE															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	SYNC1PULSE															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SYNC1PULSE															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 5.94 DC_NEXT_SYNC1_PULSE Register Contents

Bit Position	Bit Name	Function	R/W
b63 to b0	SYNC1PULSE	Indicate the system time of the next SYNC1 pulse in ns units. Accessing from EtherCAT master: Read only	R

5.4.91 DC_SYNC0_CYC_TIME — SYNC0 Cycle Time Register

This register is used to set the time between two consecutive SYNC0 pulses.

Writing to this register depends on the setting of bit 0 of the cyclic unit control register (DC_CYC_CONT at 0980h).

Address: 4401 09A0h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	SYNC0CYC															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SYNC0CYC															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 5.95 DC_SYNC0_CYC_TIME Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	SYNC0CYC	Set the time between two consecutive SYNC0 pulses in ns units. 0: Single shot mode. Only one SYNC0 pulse is generated in single shot mode. Accessing from EtherCAT master: R(/W)	R(/W)

5.4.92 DC_SYNC1_CYC_TIME — SYNC1 Cycle Time Register

This register is used to set the time between SYNC1 and SYNC0 pulses.

Writing to this register depends on the setting of bit 0 of the cyclic unit control register (DC_CYC_CONT at 0980h).

Address: 4401 09A4h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	SYNC1CYC															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SYNC1CYC															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 5.96 DC_SYNC1_CYC_TIME Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	SYNC1CYC	Set the time between SYNC1 and SYNC0 pulses in ns units. Accessing from EtherCAT master: R(/W)	R(/W)

5.4.93 DC_LATCH0_CONT — Latch0 Control Register

This register is used to control the edge function of the latch 0 input signal.

Address: 4401 09A8h

Bit	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	NEGEDGE	POSEDGE
Value after reset	X	X	X	X	X	X	0	0

Table 5.97 DC_LATCH0_CONT Register Contents

Bit Position	Bit Name	Function	R/W
b7 to b2	Reserved	When read, the value returned is undefined. Accessing from EtherCAT master: Read only	R
b1	NEGEDGE	Sets the function of the falling edge of the latch 0 input signal. 0: Continuous latch active 1: Single event (only first event active) Accessing from EtherCAT master: R(/W)	R(/W)
b0	POSEDGE	Sets the function of the rising edge of the latch 0 input signal. 0: Continuous latch active 1: Single event (only first event active) Accessing from EtherCAT master: R(/W)	R(/W)

5.4.94 DC_LATCH1_CONT — Latch1 Control Register

This register is used to control the edge function of the latch 1 input signal.

Address: 4401 09A9h

Bit	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	NEGEDGE	POSEDGE
Value after reset	X	X	X	X	X	X	0	0

Table 5.98 DC_LATCH1_CONT Register Contents

Bit Position	Bit Name	Function	R/W
b7 to b2	Reserved	When read, the value returned is undefined. Accessing from EtherCAT master: Read only	R
b1	NEGEDGE	Sets the function of the falling edge of the latch 1 input signal. 0: Continuous Latch active 1: Single event (only first event active) Accessing from EtherCAT master: R(/W)	R(/W)
b0	POSEDGE	Sets the function of the rising edge of the latch 1 input signal. 0: Continuous latch active 1: Single event (only first event active) Accessing from EtherCAT master: R(/W)	R(/W)

5.4.95 DC_LATCH0_STAT — Latch0 Status Register

This register indicates the state of the latch 0 input signal.

Address: 4401 09AEh

Bit	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	PINSTATE	EVENTNEG	EVENTPOS
Value after reset	X	X	X	X	X	0	0	0

Table 5.99 DC_LATCH0_STAT Register Contents

Bit Position	Bit Name	Function	R/W
b7 to b3	Reserved	When read, the value returned is undefined. Accessing from EtherCAT master: Read only	R
b2	PINSTATE	Indicates the state of the latch 0 input pin. Accessing from EtherCAT master: Read only	R
b1	EVENTNEG	Indicates detection of falling edges of the event latch 0 signal. 0: Falling edge not detected or continuous mode 1: Falling edge detected and mode is single-event. This flag is cleared by reading the Latch0 time negative edge register (DC_LATCH0_TIME_NEG at 09B8h). Accessing from EtherCAT master: Read only	R
b0	EVENTPOS	Indicates detection of rising edges of the event latch 0 signal. 0: Rising edge not detected or continuous mode 1: Rising edge detected and mode is single-event. This flag is cleared by reading the Latch0 time positive edge register (DC_LATCH0_TIME_POS at 09B0h). Accessing from EtherCAT master: Read only	R

5.4.96 DC_LATCH1_STAT — Latch1 Status Register

This register indicates the state of the latch 1 input signal.

Address: 4401 09AFh

Bit	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	PINSTATE	EVENTNEG	EVENTPOS
Value after reset	X	X	X	X	X	0	0	0

Table 5.100 DC_LATCH1_STAT Register Contents

Bit Position	Bit Name	Function	R/W
b7 to b3	Reserved	When read, the value returned is undefined. Accessing from EtherCAT master: Read only	R
b2	PINSTATE	Indicates the state of the latch 1 input pin. Accessing from EtherCAT master: Read only	R
b1	EVENTNEG	Indicates detection of falling edges of the event latch 1 signal. 0: Falling edge not detected or continuous mode 1: Falling edge detected and mode is single-event. This flag is cleared by reading the Latch1 time negative edge register (DC_LATCH1_TIME_NEG at 09C8h). Accessing from EtherCAT master: Read only	R
b0	EVENTPOS	Indicates detection of rising edges of the event latch 1 signal. 0: Rising edge not detected or continuous mode 1: Rising edge detected and mode is single-event. This flag is cleared by reading the Latch1 time positive edge register (DC_LATCH1_TIME_POS at 09C0h). Accessing from EtherCAT master: Read only	R

5.4.97 DC_LATCH0_TIME_POS — Latch0 Time Positive Edge Register

This register indicates the system time at the rising edge of the latch 0 input signal.

Bits 63 to 8 are internally latched (ECAT/PDI independently) when bits 7 to 0 are read, which guarantees reading a consistent value.

Clearing the latch 0 status flag function depends on the setting of bit 4 of the cyclic unit control register (DC_CYC_CONT at 0980h).

Address: 4401 09B0h

Bit	b63	b62	b61	b60	b59	b58	b57	b56	b55	b54	b53	b52	b51	b50	b49	b48
	SYSTIME															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b47	b46	b45	b44	b43	b42	b41	b40	b39	b38	b37	b36	b35	b34	b33	b32
	SYSTIME															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	SYSTIME															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SYSTIME															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 5.101 DC_LATCH0_TIME_POS Register Contents

Bit Position	Bit Name	Function	R/W
b63 to b0	SYSTIME	Indicate the system time captured at the rising edge of the latch 0 input signal. Reading this register clears bit 0 of the Latch0 status register (DC_LATCH0_STAT at 09AEh). Accessing from EtherCAT master: Read only (ack)	R (ack)

5.4.98 DC_LATCH0_TIME_NEG — Latch0 Time Negative Edge Register

This register indicates the system time at the falling edge of the latch 0 input signal.

Bits 63 to 8 are internally latched (ECAT/PDI independently) when bits 7 to 0 are read, which guarantees reading a consistent value.

Clearing the latch 0 status flag function depends on the setting of bit 4 of the cyclic unit control register (DC_CYC_CONT at 0980h).

Address: 4401 09B8h

Bit	b63	b62	b61	b60	b59	b58	b57	b56	b55	b54	b53	b52	b51	b50	b49	b48
	SYSTIME															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b47	b46	b45	b44	b43	b42	b41	b40	b39	b38	b37	b36	b35	b34	b33	b32
	SYSTIME															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	SYSTIME															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SYSTIME															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 5.102 DC_LATCH0_TIME_NEG Register Contents

Bit Position	Bit Name	Function	R/W
b63 to b0	SYSTIME	Indicate the system time captured at the falling edge of the latch 0 input signal. Reading this register clears bit 1 of the Latch0 status register (DC_LATCH0_STAT at 09AEh). Accessing from EtherCAT master: Read only (ack)	R (ack)

5.4.99 DC_LATCH1_TIME_POS — Latch1 Time Positive Edge Register

This register indicates the system time at the rising edge of the latch 1 input signal.

Bits 63 to 8 are internally latched (ECAT/PDI independently) when bits 7 to 0 are read, which guarantees reading a consistent value.

Clearing the latch 1 status flag function depends on the setting of bit 5 of the cyclic unit control register (DC_CYC_CONT at 0980h).

Address: 4401 09C0h

Bit	b63	b62	b61	b60	b59	b58	b57	b56	b55	b54	b53	b52	b51	b50	b49	b48
	SYSTIME															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b47	b46	b45	b44	b43	b42	b41	b40	b39	b38	b37	b36	b35	b34	b33	b32
	SYSTIME															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	SYSTIME															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SYSTIME															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 5.103 DC_LATCH1_TIME_POS Register Contents

Bit Position	Bit Name	Function	R/W
b63 to b0	SYSTIME	Indicate the system time captured at the rising edge of the latch 1 input signal. Reading this register clears bit 0 of the Latch1 status register (DC_LATCH1_STAT at 09AFh). Accessing from EtherCAT master: Read only (ack)	R (ack)

5.4.100 DC_LATCH1_TIME_NEG — Latch1 Time Negative Edge Register

This register indicates the system time at the falling edge of the latch 1 input signal.

Bits 63 to 8 are internally latched (ECAT/PDI independently) when bits 7 to 0 are read, which guarantees reading a consistent value.

Clearing the latch 1 status flag function depends on the setting of bit 5 of the cyclic unit control register (DC_CYC_CONT at 0980h).

Address: 4401 09C8h

Bit	b63	b62	b61	b60	b59	b58	b57	b56	b55	b54	b53	b52	b51	b50	b49	b48
	SYSTIME															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b47	b46	b45	b44	b43	b42	b41	b40	b39	b38	b37	b36	b35	b34	b33	b32
	SYSTIME															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	SYSTIME															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SYSTIME															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 5.104 DC_LATCH1_TIME_NEG Register Contents

Bit Position	Bit Name	Function	R/W
b63 to b0	SYSTIME	Indicate the system time captured at the falling edge of the latch 1 input signal. Reading this register clears bit 1 of the Latch1 status register (DC_LATCH1_STAT at 09AFh). Accessing from EtherCAT master: Read only (ack)	R (ack)

5.4.101 DC_ECAT_CNG_EV_TIME — Buffer Change Event Time Register

This register indicates the local time at the beginning of a frame which causes SyncManager to generate an EtherCAT event (switching the buffers).

Bits 31 to 8 are internally latched (ECAT/PDI independently) when bits 7 to 0 are read, which guarantees reading a consistent value.

Address: 4401 09F0h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	ECATCHANGE															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ECATCHANGE															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 5.105 DC_ECAT_CNG_EV_TIME Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	ECATCHANGE	Indicate the local time at the beginning of a frame which causes at least one SyncManager to generate an ECAT event (switching the buffers). Accessing from EtherCAT master: Read only	R

5.4.102 DC_PDI_START_EV_TIME — PDI Buffer Start Event Time Register

This register indicates the local time when SyncManager has generated a PDI event (access to the address where a buffer starts).

Bits 31 to 8 are internally latched (ECAT/PDI independently) when bits 7 to 0 are read, which guarantees reading a consistent value.

Address: 4401 09F8h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	PDISTART															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PDISTART															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 5.106 DC_PDI_START_EV_TIME Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	PDISTART	Indicate the local time when at least one SyncManager has generated a PDI event (access to the address where a buffer starts). Accessing from EtherCAT master: Read only	R

5.4.103 DC_PDI_CNG_EV_TIME — PDI Buffer Change Event Time Register

This register indicates the local time when SyncManager has generated a PDI event (switching the buffers).

Bits 31 to 8 are internally latched (ECAT/PDI independently) when bits 7 to 0 are read, which guarantees reading a consistent value.

Address: 4401 09FCh

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	PDICHANGE															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PDICHANGE															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 5.107 DC_PDI_CNG_EV_TIME Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	PDICHANGE	Indicate the local time when at least one SyncManager has generated a PDI event (switching the buffers). Accessing from EtherCAT master: Read only	R

5.4.104 PRODUCT_ID — PRODUCT ID Register

This register indicates the product ID.

Address: 4401 0E00h

Bit	b63	b62	b61	b60	b59	b58	b57	b56	b55	b54	b53	b52	b51	b50	b49	b48
	PROID															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Bit	b47	b46	b45	b44	b43	b42	b41	b40	b39	b38	b37	b36	b35	b34	b33	b32
	PROID															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	PROID															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PROID															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 5.108 PRODUCT_ID Register Contents

Bit Position	Bit Name	Function	R/W
b63 to b0	PROID	Product ID (RZ/N1D) 0x0003_0000_0000_0000 (RZ/N1S, RZ/N1L) 0x0004_0000_0000_0000 Accessing from EtherCAT master: Read only	R

5.4.105 VENDOR_ID — Vendor ID Register

This register indicates the vendor ID.

Address: 4401 0E08h

Bit	b63	b62	b61	b60	b59	b58	b57	b56	b55	b54	b53	b52	b51	b50	b49	b48
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b47	b46	b45	b44	b43	b42	b41	b40	b39	b38	b37	b36	b35	b34	b33	b32
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	VENDORID															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	VENDORID															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 5.109 VENDOR_ID Register Contents

Bit Position	Bit Name	Function	R/W
b63 to b32	Reserved	When read, the value returned is undefined. Accessing from EtherCAT master: Read only	R
b31 to b0	VENDORID	Vendor ID Accessing from EtherCAT master: Read only	R

5.4.106 USER_RAM — User RAM

This area of RAM indicates the supported features dependent on the module configuration, and takes up the 128 bits from 4401 0F80h to 4401 0FFFh. An initial value of 1 means that the implementation of this module supports the corresponding feature, except in the case of bits 7 to 0, which indicate the number of bits defined in the user RAM and the initial value is 33h for this module.

Table 5.110 User RAM Register Memory Description

Bit Position	Description	Initial Value
7 to 0	Number of bits for defining extended functionality. The value is 51 (33h) for this module.	33h
8	Upper 2 bytes of DL control register (0102h:0103h)	1
9	AL status code register (0134h:0135h)	1
10	ECAT event mask (0200h:0201h)	1
11	Configured station alias (0012h:0013h)	1
12	General input (0F18h:0F1Fh)	0
13	General output (0F10h:0F17h)	0
14	AL event mask (0204h:0207h)	1
15	Physical read/write offset (0108h:0109h)	1
16	Watchdog divider writable (0400h:0401h) and watchdog PDI (0410h:0F11h)	1
17	Watchdog counter (0442h:0443h)	1
18	Write protection (0020h:0031h)	1
19	Reset (0040h:0041h)	1
20	Reserved	0
21	DC SyncManager event time (09F0h:09FFh)	1
22	ECAT processing unit/PDI error counter (030Ch:030Dh)	1
23	EEPROM size configurable (Bit 7 at 0502h) 0: EEPROM size fixed up to 16 Kbits 1: EEPROM Size configurable	1
26 to 24	Reserved	0
27	Lost link counter (0310h:0313h)	1
28	MII management interface (0510h:0515h)	1
29	Enhanced link detection MII	1
30	Enhanced link detection EBUS	0
31	Run LED	1
32	Link/activity LED	1
33	Reserved	0
35 to 34	Reserved	1
36	Reserved	0
37	Reserved	1
38	DC Time loop control assigned to PDI	0
39	Link detection and configuration by MI	0
40	MI control by PDI	1
41	Automatic TX shift	1
42	EEPROM emulation	0
49 to 43	Reserved	0
50	ERR LED, RUN/ERR LED override	1

5.4.107 DATA_RAM — Process Data RAM

The process data RAM is used for process data and mailbox, and takes up 8 Kbytes from 4401 1000h to 4401 2FFFh. This RAM is only accessible when the EEPROM is correctly loaded (i.e., when bit 0 is 1 in the ESC DL status register, ESC_DL_STATUS, at 0110h).

5.5 Operation

5.5.1 Initializing

The initializing sequence in this section is an example used for preparation of system environments for using EtherCAT under configuration below.

Note 1. See section **RZ/N1 Initialize Sequence** in RZ/N1 User's Manual: System Introduction, Multiplexing, Electrical and Mechanical Information.

Note 2. See section **Ethernet Accessory Register**

Figure 5.2 Initializing Flowchart

Note 1. If blank EEPROM is accepted in initialization, this procedure is needed.

Figure 5.3 Initializing Flowchart (Continued)

5.6 Usage notes

5.6.1 Restriction

EtherCAT and Beckhoff ESCs have some general requirements to Ethernet PHYs.

Refer to the section "*5.1 Requirements to Ethernet PHYs*" of the EtherCAT Slave Controller IP Core (v2.04) datasheet provided by Beckhoff Automation GmbH.

MII is recommended since the PHY delay (and delay jitter) is smaller in comparison to RMII.

Since RMII PHYs include TX FIFOs, they increase the forwarding delay of an EtherCAT slave device as well as the jitter. RMII is not recommended due to these reasons.

5.6.2 Reset Circuit

Figure 5.4 shows the reset circuit of EtherCAT Slave Controller. When reset request by ESC_RESET_ECAT (0x0040) or ESC_RESET_PDI (0x0041) is received, ESC stops and reset output from ESC becomes 1. At the same time, ETHCAT_RST_Int is generated and CAT_RESETOUT_N pin outputs low level.

To release reset state of ESC, RSTN_B bit of PWRCTRL_ECAT register must be changed 1 → 0 → 1 after ETHCAT_RST_Int is detected. When reset input to ESC is deasserted, reset output from ESC becomes 0 and simultaneously ESC is rebooted with loading EEPROM. It takes around 1 ms to finish EEPROM loading. Adjust clear timing of RSTN_B bit of PWRCTRL_ECAT register so that Ethernet PHY have enough reset duration. A time chart example is shown in **Figure 5.5**.

Figure 5.4 Reset Circuit of EtherCAT Slave Controller

Figure 5.5 Reset Timing of EtherCAT Slave Controller

Section 6 Ethernet MAC 10/100/1000 (GMAC)

Portions Copyright © 2014 Synopsys. Used with permission. All rights reserved. Synopsys & DesignWare are registered trademarks of Synopsys.

6.1 Overview

- Two MAC instances (GMAC1, GMAC2)
- Compliance with the following standards:
  - IEEE 802.3-2008 for Ethernet MAC, Gigabit Media Independent Interface (GMII) & Media Independent Interface (MII)
  - IEEE 1588-2008 v2 standard for precision networked clock synchronization, IEEE 1588-2008 v2 is compliant with Power IEEE-C37.238 profile
  - IEEE 802.3-az, version D2.0 for Energy Efficient Ethernet (EEE)
- Support for 10/100/1000 Mbps data transfer rates
- Support for both half-duplex and full-duplex operation
- Programmable frame length to support both standard and “jumbo” ethernet frames with size up to 16 Kbytes (16 KB-1)
  - Jumbo mode support in cut through mode only (not implemented in store & forward due to TX & RX fifo size)
- 17 MAC Address registers for the Address Filter block
- Variety of flexible addresses filtering modes are supported
  - Perfect (DA) address filters with masks for each byte
  - SA address comparison check with masks for each byte
  - 256-bit Hash filter for multicast and unicast (DA) addresses
  - Option to pass all multicast addressed frames
  - Promiscuous mode support to pass all frames without any filtering for network monitoring
  - Passes all incoming packets (as per filter) with a status report.
- Native DMA with simple independent channels Transmit and Receive engines
  - 1 RX channels, FIFO’s size 4 KB for receive channel
  - 1 TX channels, FIFO’s size 2 KB for transmit channel
  - DMA implements dual buffer (ring) or linked list (chained) descriptor chaining
- Advanced IEEE 1588-2002 & 2008 ethernet frame time stamping support.
  - IEEE1588 Time base information, with reference clock of 25 MHz or 125 MHz (same sources as GMII)
  - IEEE1588 external snapshot
- Provides the flexibility to control the Pulse Per Second (PPS) output signal (GMAC1 only)
- Programmable CRC generation and checking
- Support for RMON statistics (L2 layer only)
- Station Management Block, MDIO interface

- Ethernet Energy Efficiency compliant with IEEE 802.3az-2010
  - Support for wake up on LAN on magic packet and packet filtering.
  - Support Energy Efficiency feature (LPI Mode)
  - Wake-up capability
- Interface
  - Native mode GMII (used in internal only), MII
  - No Native RMII, RGMII, managed by RMII/RGMII convertor (connected on external pins)

Figure 6.1 GMAC Block Diagram

6.2 Signal Interface

Table 6.1 Signal Interface of the GMAC (excluding PHY MII pins)

Signal Name	I/O	Description	Active
Clock			
GMAC[m]_HCLK	I	AHB clock used for the register interface	
GMAC[m]_XCLK	I	AXI clock	
Interrupt			
GMAC[m]_SBD_Int	O	GMAC[m] general, level sensitive	High
GMAC[m]_LPI_Int	O	GMAC[m] energy efficient, level sensitive	High
GMAC[m]_PMT_Int	O	GMAC[m] power management, level sensitive	High
Signals for PTP			
GMAC_PTP_REFCLK_I	I	Reference clock for PTP	
GMAC1_PTP_TIMESTAMP_O[63:0]	O	Timestamp value output for PTP	
GMAC2_PTP_TIMESTAMP_I[63:0]	I	Timestamp value input for PTP	
External Signal			
MAC_PPS[0]	O	Pulse Per Second output. (PPS0 of GMAC1)	High
MAC_PPS[1]	O	Pulse Per Second output. (PPS1 of GMAC1)	High
MAC_TRIG[1]	I	Auxiliary Timestamp Trigger Input (GMAC1)	Rise
MAC_TRIG[2]	I	Auxiliary Timestamp Trigger Input (GMAC2)	Rise

Note: m = 1 or 2

6.3 Register Map

6.3.1 GMAC1 Register Map

Table 6.2 GMAC1 Register Map (1/3)

Address	Register Symbol	Register Name
4400 0000h	MAC_Configuration	MAC Configuration Register
4400 0004h	MAC_Frame_Filter	MAC Frame Filter Register
4400 0010h	GMII_Address	GMII Address Register
4400 0014h	GMII_Data	GMII Data Register
4400 0018h	Flow_Control	Flow Control Register
4400 001Ch	VLAN_Tag	VLAN Tag Register
4400 0020h	Version	Version Register
4400 0024h	Debug	Debug Register
4400 0028h	Remote_Wake_Up_Frame_Filter	Remote Wake-Up Frame Filter Register
4400 002Ch	PMT_Control_Status	PMT Control and Status Register
4400 0030h	LPI_Control_Status	LPI Control and Status Register
4400 0034h	LPI_Timers_Control	LPI Timers Control Register
4400 0038h	Interrupt_Status	Interrupt Status Register
4400 003Ch	Interrupt_Mask	Interrupt Mask Register
4400 0040h + 8h × n	MAC_Address[n]_High (n = 0..15)	MAC Address[n] High Register
4400 0044h + 8h × n	MAC_Address[n]_Low (n = 0..15)	MAC Address[n] Low Register
4400 00DCh	WDog_Timeout	Watchdog Timeout Register
4400 0100h	MMC_Control	MMC Control Register
4400 0104h	MMC_Receive_Interrupt	MMC Receive Interrupt Register
4400 0108h	MMC_Transmit_Interrupt	MMC Transmit Interrupt Register
4400 010Ch	MMC_Receive_Interrupt_Mask	MMC Receive Interrupt Mask Register
4400 0110h	MMC_Transmit_Interrupt_Mask	MMC Transmit Interrupt Mask Register
4400 0114h	Tx_Octet_Count_Good_Bad	Transmit Octet Count for Good and Bad Frames
4400 0118h	Tx_Frame_Count_Good_Bad	Transmit Frame Count for Good and Bad Frames
4400 011Ch	Tx_Broadcast_Frames_Good	Transmit Frame Count for Good Broadcast Frames
4400 0120h	Tx_Multicast_Frames_Good	Transmit Frame Count for Good Multicast Frames
4400 0124h	Tx_64Octets_Frames_Good_Bad	Transmit Octet Count for Good and Bad 64 Byte Frames
4400 0128h	Tx_65To127Octets_Frames_Good_Bad	Transmit Octet Count for Good and Bad 65 to 127 Bytes Frames
4400 012Ch	Tx_128To255Octets_Frames_Good_Bad	Transmit Octet Count for Good and Bad 128 to 255 Bytes Frames
4400 0130h	Tx_256To511Octets_Frames_Good_Bad	Transmit Octet Count for Good and Bad 256 to 511 Bytes Frames
4400 0134h	Tx_512To1023Octets_Frames_Good_Bad	Transmit Octet Count for Good and Bad 512 to 1023 Bytes Frames
4400 0138h	Tx_1024ToMaxOctets_Frames_Good_Bad	Transmit Octet Count for Good and Bad 1024 to Maxsize Bytes Frames
4400 013Ch	Tx_Unicast_Frames_Good_Bad	Transmit Frame Count for Good and Bad Unicast Frames
4400 0140h	Tx_Multicast_Frames_Good_Bad	Transmit Frame Count for Good and Bad Multicast Frames
4400 0144h	Tx_Broadcast_Frames_Good_Bad	Transmit Frame Count for Good and Bad Broadcast Frames
4400 0148h	Tx_Underflow_Error_Frames	Transmit Frame Count for Underflow Error Frames
4400 014Ch	Tx_Single_Collision_Good_Frames	Transmit Frame Count for Frames Transmitted after Single Collision

Table 6.2 GMAC1 Register Map (2/3)

Address	Register Symbol	Register Name
4400 0150h	Tx_Multiple_Collision_Good_Frames	Transmit Frame Count for Frames Transmitted after Multiple Collision
4400 0154h	Tx_Deferred_Frames	Transmit Frame Count for Deferred Frames
4400 0158h	Tx_Late_Collision_Frames	Transmit Frame Count for Late Collision Error Frames
4400 015Ch	Tx_Excessive_Collision_Frames	Transmit Frame Count for Excessive Collision Error Frames
4400 0160h	Tx_Carrier_Error_Frames	Transmit Frame Count for Carrier Sense Error Frames
4400 0164h	Tx_Octet_Count_Good	Transmit Octet Count for Good Frames
4400 0168h	Tx_Frame_Count_Good	Transmit Frame Count for Good Frames
4400 016Ch	Tx_Excessive_Deferral_Error	Transmit Frame Count for Excessive Deferral Error Frames
4400 0170h	Tx_Pause_Frames	Transmit Frame Count for Good PAUSE Frames
4400 0174h	Tx_VLAN_Frames_Good	Transmit Frame Count for Good VLAN Frames
4400 0178h	Tx_OSize_Frames_Good	Transmit Frame Count for Good Oversize Frames
4400 0180h	Rx_Frames_Count_Good_Bad	Receive Frame Count for Good and Bad Frames
4400 0184h	Rx_Octet_Count_Good_Bad	Receive Octet Count for Good and Bad Frames
4400 0188h	Rx_Octet_Count_Good	Receive Octet Count for Good Frames
4400 018Ch	Rx_Broadcast_Frames_Good	Receive Frame Count for Good Broadcast Frames
4400 0190h	Rx_Multicast_Frames_Good	Receive Frame Count for Good Multicast Frames
4400 0194h	Rx_CRC_Error_Frames	Receive Frame Count for CRC Error Frames
4400 0198h	Rx_Alignment_Error_Frames	Receive Frame Count for Alignment Error Frames
4400 019Ch	Rx_Runt_Error_Frames	Receive Frame Count for Runt Error Frames
4400 01A0h	Rx_Jabber_Error_Frames	Receive Frame Count for Jabber Error Frames
4400 01A4h	Rx_Undersize_Frames_Good	Receive Frame Count for Undersize Frames
4400 01A8h	Rx_Oversize_Frames_Good	Receive Frame Count for Oversize Frames
4400 01ACh	Rx_64Octets_Frames_Good_Bad	Receive Frame Count for Good and Bad 64 Byte Frames
4400 01B0h	Rx_65To127Octets_Frames_Good_Bad	Receive Frame Count for Good and Bad 65 to 127 Bytes Frames
4400 01B4h	Rx_128To255Octets_Frames_Good_Bad	Receive Frame Count for Good and Bad 128 to 255 Bytes Frames
4400 01B8h	Rx_256To511Octets_Frames_Good_Bad	Receive Frame Count for Good and Bad 256 to 511 Bytes Frames
4400 01BCh	Rx_512To1023Octets_Frames_Good_Bad	Receive Frame Count for Good and Bad 512 to 1,023 Bytes Frames
4400 01C0h	Rx_1024ToMaxOctets_Frames_Good_Bad	Receive Frame Count for Good and Bad 1,024 to Maxsize Bytes Frames
4400 01C4h	Rx_Unicast_Frames_Good	Receive Frame Count for Good Unicast Frames
4400 01C8h	Rx_Length_Error_Frames	Receive Frame Count for Length Error Frames
4400 01CCh	Rx_Out_Of_Range_Type_Frames	Receive Frame Count for Out of Range Frames
4400 01D0h	Rx_Pause_Frames	Receive Frame Count for PAUSE Frames
4400 01D4h	Rx_FIFO_Overflow_Frames	Receive Frame Count for FIFO Overflow Frames
4400 01D8h	Rx_VLAN_Frames_Good_Bad	Receive Frame Count for Good and Bad VLAN Frames
4400 01DCh	Rx_Watchdog_Error_Frames	Receive Frame Count for Watchdog Error Frames
4400 01E0h	Rx_Receive_Error_Frames	Receive Frame Count for Receive Error Frames
4400 01E4h	Rx_Control_Frames_Good	Receive Frame Count for Good Control Frames
4400 0500h + 4h × n	Hash_Table_Reg[n] (n = 0..7)	Hash Table Register [n]
4400 0588h	VLAN_Hash_Table_Reg	VLAN Hash Table Register
4400 0700h	Timestamp_Control	Timestamp Control Register
4400 0704h	Sub_Second_Increment	Sub-Second Increment Register

Table 6.2 GMAC1 Register Map (3/3)

Address	Register Symbol	Register Name
4400 0708h	System_Time_Seconds	System Time - Seconds Register
4400 070Ch	System_Time_Nanoseconds	System Time - Nanoseconds Register
4400 0710h	System_Time_Seconds_Update	System Time - Seconds Update Register
4400 0714h	System_Time_Nanoseconds_Update	System Time - Nanoseconds Update Register
4400 0718h	Timestamp_Addend	Timestamp Addend Register
4400 071Ch	Target_Time_Seconds	Target Time Seconds Register
4400 0720h	Target_Time_Nanoseconds	Target Time Nanoseconds Register
4400 0728h	Timestamp_Status	Timestamp Status Register
4400 072Ch	PPS_Control	PPS Control Register
4400 0730h	Auxiliary_Timestamp_Nanoseconds	Auxiliary Timestamp - Nanoseconds Register
4400 0734h	Auxiliary_Timestamp_Seconds	Auxiliary Timestamp - Seconds Register
4400 0760h	PPS0_Interval	PPS0 Interval Register
4400 0764h	PPS0_Width	PPS0 Width Register
4400 0780h	PPS1_Target_Time_Seconds	PPS1 Target Time Seconds Register
4400 0784h	PPS1_Target_Time_Nanoseconds	PPS1 Target Time Nanoseconds Register
4400 0788h	PPS1_Interval	PPS1 Interval Register
4400 078Ch	PPS1_Width	PPS1 Width Register
4400 0800h	MAC_Address16_High	MAC Address16 High Register
4400 0804h	MAC_Address16_Low	MAC Address16 Low Register
4400 0808h	MAC_Address17_High	MAC Address17 High Register
4400 080Ch	MAC_Address17_Low	MAC Address17 Low Register
4400 1000h	Bus_Mode	Bus Mode Register
4400 1004h	Transmit_Poll_Demand	Transmit Poll Demand Register
4400 1008h	Receive_Poll_Demand	Receive Poll Demand Register
4400 100Ch	Receive_Descriptor_List_Address	Receive Descriptor List Address Register
4400 1010h	Transmit_Descriptor_List_Address	Transmit Descriptor List Address Register
4400 1014h	Status	Status Register
4400 1018h	Operation_Mode	Operation Mode Register
4400 101Ch	Interrupt_Enable	Interrupt Enable Register
4400 1020h	Missed_Frame_And_Buffer_Overflow_Counter	Missed Frame and Buffer Overflow Counter Register
4400 1024h	Receive_Interrupt_Watchdog_Timer	Receive Interrupt Watchdog Timer Register
4400 1028h	AXI_Bus_Mode	AXI Bus Mode Register
4400 102Ch	AXI_Status	AXI Status Register
4400 1048h	Current_Host_Transmit_Descriptor	Current Host Transmit Descriptor Register
4400 104Ch	Current_Host_Receive_Descriptor	Current Host Receive Descriptor Register
4400 1050h	Current_Host_Transmit_Buffer_Address	Current Host Transmit Buffer Address Register
4400 1054h	Current_Host_Receive_Buffer_Address	Current Host Receive Buffer Address Register
4400 1058h	HW_Feature	HW Feature Register

6.3.2 GMAC2 Register Map

Table 6.3 GMAC2 Register Map (1/3)

Address	Register Symbol	Register Name
4400 2000h	MAC_Configuration	MAC Configuration Register
4400 2004h	MAC_Frame_Filter	MAC Frame Filter Register
4400 2010h	GMII_Address	GMII Address Register
4400 2014h	GMII_Data	GMII Data Register
4400 2018h	Flow_Control	Flow Control Register
4400 201Ch	VLAN_Tag	VLAN Tag Register
4400 2020h	Version	Version Register
4400 2024h	Debug	Debug Register
4400 2028h	Remote_Wake_Up_Frame_Filter	Remote Wake-Up Frame Filter Register
4400 202Ch	PMT_Control_Status	PMT Control and Status Register
4400 2030h	LPI_Control_Status	LPI Control and Status Register
4400 2034h	LPI_Timers_Control	LPI Timers Control Register
4400 2038h	Interrupt_Status	Interrupt Status Register
4400 203Ch	Interrupt_Mask	Interrupt Mask Register
4400 2040h + 8h × n	MAC_Address[n]_High (n = 0..15)	MAC Address[n] High Register
4400 2044h + 8h × n	MAC_Address[n]_Low (n = 0..15)	MAC Address[n] Low Register
4400 20DCh	WDog_Timeout	Watchdog Timeout Register
4400 2100h	MMC_Control	MMC Control Register
4400 2104h	MMC_Receive_Interrupt	MMC Receive Interrupt Register
4400 2108h	MMC_Transmit_Interrupt	MMC Transmit Interrupt Register
4400 210Ch	MMC_Receive_Interrupt_Mask	MMC Receive Interrupt Mask Register
4400 2110h	MMC_Transmit_Interrupt_Mask	MMC Transmit Interrupt Mask Register
4400 2114h	Tx_Octet_Count_Good_Bad	Transmit Octet Count for Good and Bad Frames
4400 2118h	Tx_Frame_Count_Good_Bad	Transmit Frame Count for Good and Bad Frames
4400 211Ch	Tx_Broadcast_Frames_Good	Transmit Frame Count for Good Broadcast Frames
4400 2120h	Tx_Multicast_Frames_Good	Transmit Frame Count for Good Multicast Frames
4400 2124h	Tx_64Octets_Frames_Good_Bad	Transmit Octet Count for Good and Bad 64 Byte Frames
4400 2128h	Tx_65To127Octets_Frames_Good_Bad	Transmit Octet Count for Good and Bad 65 to 127 Bytes Frames
4400 212Ch	Tx_128To255Octets_Frames_Good_Bad	Transmit Octet Count for Good and Bad 128 to 255 Bytes Frames
4400 2130h	Tx_256To511Octets_Frames_Good_Bad	Transmit Octet Count for Good and Bad 256 to 511 Bytes Frames
4400 2134h	Tx_512To1023Octets_Frames_Good_Bad	Transmit Octet Count for Good and Bad 512 to 1023 Bytes Frames
4400 2138h	Tx_1024ToMaxOctets_Frames_Good_Bad	Transmit Octet Count for Good and Bad 1024 to Maxsize Bytes Frames
4400 213Ch	Tx_Unicast_Frames_Good_Bad	Transmit Frame Count for Good and Bad Unicast Frames
4400 2140h	Tx_Multicast_Frames_Good_Bad	Transmit Frame Count for Good and Bad Multicast Frames
4400 2144h	Tx_Broadcast_Frames_Good_Bad	Transmit Frame Count for Good and Bad Broadcast Frames
4400 2148h	Tx_Underflow_Error_Frames	Transmit Frame Count for Underflow Error Frames
4400 214Ch	Tx_Single_Collision_Good_Frames	Transmit Frame Count for Frames Transmitted after Single Collision
4400 2150h	Tx_Multiple_Collision_Good_Frames	Transmit Frame Count for Frames Transmitted after Multiple Collision
4400 2154h	Tx_Deferred_Frames	Transmit Frame Count for Deferred Frames

Table 6.3 GMAC2 Register Map (2/3)

Address	Register Symbol	Register Name
4400 2158h	Tx_Late_Collision_Frames	Transmit Frame Count for Late Collision Error Frames
4400 215Ch	Tx_Excessive_Collision_Frames	Transmit Frame Count for Excessive Collision Error Frames
4400 2160h	Tx_Carrier_Error_Frames	Transmit Frame Count for Carrier Sense Error Frames
4400 2164h	Tx_Octet_Count_Good	Transmit Octet Count for Good Frames
4400 2168h	Tx_Frame_Count_Good	Transmit Frame Count for Good Frames
4400 216Ch	Tx_Excessive_Deferral_Error	Transmit Frame Count for Excessive Deferral Error Frames
4400 2170h	Tx_Pause_Frames	Transmit Frame Count for Good PAUSE Frames
4400 2174h	Tx_VLAN_Frames_Good	Transmit Frame Count for Good VLAN Frames
4400 2178h	Tx_OSize_Frames_Good	Transmit Frame Count for Good Oversize Frames
4400 2180h	Rx_Frames_Count_Good_Bad	Receive Frame Count for Good and Bad Frames
4400 2184h	Rx_Octet_Count_Good_Bad	Receive Octet Count for Good and Bad Frames
4400 2188h	Rx_Octet_Count_Good	Receive Octet Count for Good Frames
4400 218Ch	Rx_Broadcast_Frames_Good	Receive Frame Count for Good Broadcast Frames
4400 2190h	Rx_Multicast_Frames_Good	Receive Frame Count for Good Multicast Frames
4400 2194h	Rx_CRC_Error_Frames	Receive Frame Count for CRC Error Frames
4400 2198h	Rx_Alignment_Error_Frames	Receive Frame Count for Alignment Error Frames
4400 219Ch	Rx_Runt_Error_Frames	Receive Frame Count for Runt Error Frames
4400 21A0h	Rx_Jabber_Error_Frames	Receive Frame Count for Jabber Error Frames
4400 21A4h	Rx_Undersize_Frames_Good	Receive Frame Count for Undersize Frames
4400 21A8h	Rx_Oversize_Frames_Good	Receive Frame Count for Oversize Frames
4400 21ACh	Rx_64Octets_Frames_Good_Bad	Receive Frame Count for Good and Bad 64 Byte Frames
4400 21B0h	Rx_65To127Octets_Frames_Good_Bad	Receive Frame Count for Good and Bad 65 to 127 Bytes Frames
4400 21B4h	Rx_128To255Octets_Frames_Good_Bad	Receive Frame Count for Good and Bad 128 to 255 Bytes Frames
4400 21B8h	Rx_256To511Octets_Frames_Good_Bad	Receive Frame Count for Good and Bad 256 to 511 Bytes Frames
4400 21BCh	Rx_512To1023Octets_Frames_Good_Bad	Receive Frame Count for Good and Bad 512 to 1,023 Bytes Frames
4400 21C0h	Rx_1024ToMaxOctets_Frames_Good_Bad	Receive Frame Count for Good and Bad 1,024 to Maxsize Bytes Frames
4400 21C4h	Rx_Unicast_Frames_Good	Receive Frame Count for Good Unicast Frames
4400 21C8h	Rx_Length_Error_Frames	Receive Frame Count for Length Error Frames
4400 21CCh	Rx_Out_Of_Range_Type_Frames	Receive Frame Count for Out of Range Frames
4400 21D0h	Rx_Pause_Frames	Receive Frame Count for PAUSE Frames
4400 21D4h	Rx_FIFO_Overflow_Frames	Receive Frame Count for FIFO Overflow Frames
4400 21D8h	Rx_VLAN_Frames_Good_Bad	Receive Frame Count for Good and Bad VLAN Frames
4400 21DCh	Rx_Watchdog_Error_Frames	Receive Frame Count for Watchdog Error Frames
4400 21E0h	Rx_Receive_Error_Frames	Receive Frame Count for Receive Error Frames
4400 21E4h	Rx_Control_Frames_Good	Receive Frame Count for Good Control Frames
4400 2500h+ 4h × n	Hash_Table_Reg[n] (n = 0..7)	Hash Table Register [n]
4400 2588h	VLAN_Hash_Table_Reg	VLAN Hash Table Register
4400 2700h	Timestamp_Control	Timestamp Control Register
4400 2728h	Timestamp_Status	Timestamp Status Register
4400 2730h	Auxiliary_Timestamp_Nanoseconds	Auxiliary Timestamp - Nanoseconds Register
4400 2734h	Auxiliary_Timestamp_Seconds	Auxiliary Timestamp - Seconds Register

Table 6.3 GMAC2 Register Map (3/3)

Address	Register Symbol	Register Name
4400 2800h	MAC_Address16_High	MAC Address16 High Register
4400 2804h	MAC_Address16_Low	MAC Address16 Low Register
4400 2808h	MAC_Address17_High	MAC Address17 High Register
4400 280Ch	MAC_Address17_Low	MAC Address17 Low Register
4400 3000h	Bus_Mode	Bus Mode Register
4400 3004h	Transmit_Poll_Demand	Transmit Poll Demand Register
4400 3008h	Receive_Poll_Demand	Receive Poll Demand Register
4400 300Ch	Receive_Descriptor_List_Address	Receive Descriptor List Address Register
4400 3010h	Transmit_Descriptor_List_Address	Transmit Descriptor List Address Register
4400 3014h	Status	Status Register
4400 3018h	Operation_Mode	Operation Mode Register
4400 301Ch	Interrupt_Enable	Interrupt Enable Register
4400 3020h	Missed_Frame_And_Buffer_Overflow_Counter	Missed Frame and Buffer Overflow Counter Register
4400 3024h	Receive_Interrupt_Watchdog_Timer	Receive Interrupt Watchdog Timer Register
4400 3028h	AXI_Bus_Mode	AXI Bus Mode Register
4400 302Ch	AXI_Status	AXI Status Register
4400 3048h	Current_Host_Transmit_Descriptor	Current Host Transmit Descriptor Register
4400 304Ch	Current_Host_Receive_Descriptor	Current Host Receive Descriptor Register
4400 3050h	Current_Host_Transmit_Buffer_Address	Current Host Transmit Buffer Address Register
4400 3054h	Current_Host_Receive_Buffer_Address	Current Host Receive Buffer Address Register
4400 3058h	HW_Feature	HW Feature Register

6.4 Register Description

6.4.1 MAC_Configuration — MAC Configuration Register

Address: GMAC1: 4400 0000h
GMAC2: 4400 2000h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	TWOKPE	—	CST	—	WD	JD	BE	JE	IFG			DCRS
Value after reset	X	X	X	X	0	X	0	X	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PS	FES	DO	LM	DM	IPC	DR	—	ACS	BL		DC	TE	RE	PRELEN	
Value after reset	0	0	0	0	0	0	0	X	0	0	0	0	0	0	0	0

Table 6.4 MAC_Configuration Register Contents (1/3)

Bit Position	Bit Name	Function	R/W
b31 to b28	Reserved	Reserved	R
b27	TWOKPE	IEEE 802.3 as Support for 2 K Packets. When set, the GMAC considers all frames, with up to 2,000 bytes length, as normal packets. When Bit 20 (JE) is not set, the GMAC considers all received frames of size more than 2 Kbytes as Giant frames. When this bit is reset and Bit 20 (JE) is not set, the GMAC considers all received frames of size more than 1,518 bytes (1,522 bytes for tagged) as Giant frames. When Bit 20 is set, setting this bit has no effect on Giant Frame status.	R/W
b26	Reserved	Reserved	R
b25	CST	CRC Stripping for Type Frames When this bit is set, the last 4 bytes (FCS) of all frames of Ether type (Length/Type field greater than or equal to 1,536) are stripped and dropped before forwarding the frame to the application.	R/W
b24	Reserved	Reserved	R
b23	WD	Watchdog Disable When this bit is set, the GMAC disables the watchdog timer on the receiver. The GMAC can receive frames of up to 16,384 bytes. When this bit is reset, the GMAC does not allow a receive frame which more than 2,048 bytes (10,240 if JE is set high) or the value programmed in Watchdog Timeout Register (WDog_Timeout). The GMAC cuts off any bytes received after the watchdog limit number of bytes.	R/W
b22	JD	Jabber Disable When this bit is set, the GMAC disables the jabber timer on the transmitter. The GMAC can transfer 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 JE is set high) during transmission.	R/W
b21	BE	Frame Burst Enable When this bit is set, the GMAC allows frame bursting during transmission in the GMII half-duplex mode.	R/W
b20	JE	Jumbo Frame Enable When this bit is set, the GMAC allows Jumbo frames of 9,018 bytes (9,022 bytes for VLAN tagged frames) without reporting a giant frame error in the receive frame status.	R/W

Table 6.4 MAC_Configuration Register Contents (2/3)

Bit Position	Bit Name	Function	R/W
b19 to b17	IFG	Inter-Frame Gap These bits control the minimum IFG between frames during transmission. 3'b000: 96-bit times 3'b001: 88-bit times 3'b010: 80-bit times : 3'b111: 40-bit times In the half-duplex mode, the minimum IFG can be configured only for 64-bit times (IFG = 3'b100). Lower values are not considered. In the 1000-Mbps mode, the minimum IFG supported is 80-bit times (and above). When a JAM pattern is being transmitted because of backpressure activation, the GMAC does not consider the minimum IFG.	R/W
b16	DCRS	Disable Carrier Sense During Transmission When set high, this bit makes the GMAC transmitter ignore the (G)MII CRS signal during frame transmission in the half-duplex mode. This request results in no errors generated because of Loss of Carrier or No Carrier during such transmission. When this bit is low, the GMAC transmitter generates such errors because of Carrier Sense and can even abort the transmissions.	R/W
b15	PS	Port Select This bit selects the Ethernet line speed: 0: For 1000 Mbps operations 1: For 10 or 100 Mbps operations In 10 or 100 Mbps operations, this bit, along with FES bit, selects the exact line speed.	R/W
b14	FES	Speed This bit selects the speed in the MII interface. 0: 10 Mbps 1: 100 Mbps	R/W
b13	DO	Disable Receive Own When this bit is set, the GMAC disables the reception of frames when the Transmit Data Enable is asserted in the half-duplex mode. When this bit is reset, the GMAC receives all packets that are given by the PHY while transmitting. This bit is not applicable if the GMAC is operating in the full-duplex mode.	R/W
b12	LM	Loopback Mode When this bit is set, the GMAC operates in the loopback mode at GMII or MII. The Receive clock is required for the loopback to work properly, because the Transmit clock is not looped-back internally.	R/W
b11	DM	Duplex Mode When this bit is set, the GMAC operates in the full-duplex mode where it can transmit and receive simultaneously.	R/W
b10	IPC	Checksum Offload When this bit is set, the GMAC calculates the 16-bit one's complement of the one's complement sum of all received Ethernet frame payloads. It also checks whether the IPv4 Header checksum (assumed to be bytes 25-26 or 29-30 (VLAN-tagged) of the received Ethernet frame) is correct for the received frame and gives the status in the receive status word. When this bit is reset, this function is disabled. When this bit is set, enables the IPv4 header checksum checking and IPv4 or IPv6 TCP, UDP, or ICMP payload checksum checking. When this bit is reset, the COE function in the receiver is disabled and the corresponding Payload Checksum Error and IP Header Checksum Error status bits are always cleared.	R/W
b9	DR	Disable Retry When this bit is set, the GMAC attempts only one transmission. When a collision occurs on the GMII or MII interface, the GMAC ignores the current frame transmission and reports a Frame Abort with excessive collision error in the transmit frame status. When this bit is reset, the GMAC attempts retries based on the settings of the BL field (Bits[6:5]). This bit is applicable only in the half-duplex mode.	R/W

Table 6.4 MAC_Configuration Register Contents (3/3)

Bit Position	Bit Name	Function	R/W
b8	Reserved	Reserved	R
b7	ACS	Automatic Pad or CRC Stripping When this bit is set, the GMAC strips the Pad or FCS field on the incoming frames only if the value of the length field is less than 1,536 bytes. All received frames with length field greater than or equal to 1,536 bytes are passed to the application without stripping the Pad or FCS field. When this bit is reset, the GMAC passes all incoming frames, without modifying them, to the Host.	R/W
b6, b5	BL	Back-Off Limit The Back-Off limit determines the random integer number (r) of slot time delays (4,096-bit times for 1000 Mbps and 512-bit times for 10/100 Mbps) for which the GMAC waits before rescheduling a transmission attempt during retries after a collision. This bit is applicable only in the half-duplex mode. 2'b00: k = min (n, 10) 2'b01: k = min (n, 8) 2'b10: k = min (n, 4) 2'b11: k = min (n, 1) where n = retransmission attempt. The random integer r takes the value in the range 0 ≤ r < kth power of 2	R/W
b4	DC	Deferral Check When this bit is set, the deferral check function is enabled in the GMAC. The GMAC issues 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 the 10 or 100 Mbps mode. If the GMAC is configured for 1000 Mbps operation or if the Jumbo frame mode is enabled in the 10 or 100 Mbps mode, the threshold for deferral is 155,680 bits times. Deferral begins when the transmitter is ready to transmit, but it is prevented because of an active carrier sense signal (CRS) on GMII or MII. The defer time is not cumulative. For example, if the transmitter defers for 10,000 bit times because the CRS signal is active and then the CRS signal becomes inactive, the transmitter transmits and collision happens. Because of collision, the transmitter needs to back off and then defer again after back off completion. In such a scenario, the deferral timer is reset to 0 and it is restarted. 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 the half-duplex mode.	R/W
b3	TE	Transmitter Enable When this bit is set, the transmit state machine of the GMAC is enabled for transmission on the GMII or MII. When this bit is reset, the GMAC transmit state machine is disabled after the completion of the transmission of the current frame, and does not transmit any further frames.	R/W
b2	RE	Receiver Enable When this bit is set, the receive state machine of the GMAC is enabled for receiving frames from the GMII or MII. When this bit is reset, the GMAC receive state machine is disabled after the completion of the reception of the current frame, and does not receive any further frames from the GMII or MII.	R/W
b1, b0	PRELEN	Preamble Length for Transmit Frames These bits control the number of preamble bytes that are added to the beginning of every Transmit frame. The preamble reduction occurs only when the GMAC is operating in the full-duplex mode. 2'b00: 7 bytes of preamble 2'b01: 5 bytes of preamble 2'b10: 3 bytes of preamble 2'b11: Reserved	R/W

6.4.2 MAC_Frame_Filter — MAC Frame Filter Register

Address: GMAC1: 4400 0004h
GMAC2: 4400 2004h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	RA	—	—	—	—	—	—	—	—	—	—	—	—	—	—	VTFE
Value after reset	0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	HPF	SAF	SAIF	PCF		DBF	PM	DAIF	HMC	HUC	PR
Value after reset	X	X	X	X	X	0	0	0	0	0	0	0	0	0	0	0

Table 6.5 MAC_Frame_Filter Register Contents (1/2)

Bit Position	Bit Name	Function	R/W
b31	RA	Receive All When this bit is set, the GMAC Receiver module passes all received frames, irrespective of whether they pass the address filter or not, to the Application. The result of the SA or DA filtering is updated (pass or fail) in the corresponding bits in the Receive Status Word. When this bit is reset, the Receiver module passes only those frames to the Application that pass the SA or DA address filter.	R/W
b30 to b17	Reserved	Reserved	R
b16	VTFE	VLAN Tag Filter Enable When set, this bit enables the GMAC to drop VLAN tagged frames that do not match the VLAN Tag comparison. When reset, the GMAC forwards all frames irrespective of the match status of the VLAN Tag.	R/W
b15 to b11	Reserved	Reserved	R
b10	HPF	Hash or Perfect Filter When this bit is set, it configures the address filter to pass a frame if it matches either the perfect filtering or the hash filtering as set by the HMC or HUC bits. When this bit is low and the HUC or HMC bit is set, the frame is passed only if it matches the Hash filter.	R/W
b9	SAF	Source Address Filter Enable When this bit is set, the GMAC compares the SA field of the received frames with the values programmed in the enabled SA registers. If the comparison fails, the GMAC drops the frame. When this bit is reset, the GMAC forwards the received frame to the application with updated SAF bit of the Rx Status depending on the SA address comparison. ***Note)*** According to the IEEE specification, Bit 47 of the SA is reserved and set to 0. However, the GMAC compares all 48 bits. The software driver should take this into consideration while programming the MAC address registers for SA.	R/W
b8	SAIF	SA Inverse Filtering 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 are marked as failing the SA Address filter. When this bit is reset, frames whose SA does not match the SA registers are marked as failing the SA Address filter.	R/W

Table 6.5 MAC_Frame_Filter Register Contents (2/2)

Bit Position	Bit Name	Function	R/W
b7, b6	PCF	Pass Control Frames These bits control the forwarding of all control frames (including unicast and multicast PAUSE frames). 2'b00: GMAC filters all control frames from reaching the application. 2'b01: GMAC forwards all control frames except PAUSE control frames to application even if they fail the Address filter. 2'b10: GMAC forwards all control frames to application even if they fail the Address Filter. 2'b11: 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 Bit 2 (RFE) of Flow Control Register (Flow_Control) 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 Flow Control Register (Flow_Control) 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 2'b01 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 the full-duplex mode and flow control is not enabled, you should set the PCF field to 2'b10 or 2'b11 (as required by the application).	R/W
b5	DBF	Disable Broadcast Frames When this bit is set, the AFM module filters all incoming broadcast frames. In addition, it overrides all other filter settings. When this bit is reset, the AFM module passes all received broadcast frames.	R/W
b4	PM	Pass All Multicast When set, this bit indicates that all received frames with a multicast destination address (first bit in the destination address field is "1") are passed. When reset, filtering of multicast frame depends on HMC bit.	R/W
b3	DAIF	DA Inverse Filtering When this bit is set, the Address Check block operates in inverse filtering mode for the DA address comparison for both unicast and multicast frames. When reset, normal filtering of frames is performed.	R/W
b2	HMC	Hash Multicast When set, GMAC performs destination address filtering of received multicast frames according to the hash table. When reset, 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.	R/W
b1	HUC	Hash Unicast When set, GMAC performs destination address filtering of unicast frames according to the hash table. When reset, 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.	R/W
b0	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 or DA Filter Fails status bits of the Receive Status Word are always cleared when PR is set.	R/W

6.4.3 GMII_Address — GMII Address Register

Address: GMAC1: 4400 0010h
GMAC2: 4400 2010h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PA					GR					CR				GW	GB
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6.6 GMII_Address Register Contents (1/2)

Bit Position	Bit Name	Function	R/W
b31 to b16	Reserved	Reserved	R
b15 to b11	PA	Physical Layer Address This field indicates which of the 32 possible PHY devices are being accessed.	R/W
b10 to b6	GR	GMII Register These bits select the desired GMII register in the selected PHY device.	R/W
b5 to b2	CR	CSR Clock Range The CSR Clock (GMAC[m]_HCLK) Range selection determines the frequency of the MDC clock according to the CSR clock frequency. The suggested range of CSR clock frequency applicable for each value (when Bit[5] = 0) ensures that the MDC clock is approximately between the frequency range 1.0 MHz to 2.5 MHz. 4’b0000: The frequency of the CSR clock is 60-100 MHz, and the MDC clock is CSR clock/42. 4’b0001: The frequency of the CSR clock is 100-150 MHz, and the MDC clock is CSR clock/62. 4’b0010: The frequency of the CSR clock is 20-35 MHz, and the MDC clock is CSR clock/16. 4’b0011: The frequency of the CSR clock is 35-60 MHz, and the MDC clock is CSR clock/26. 4’b0100: The frequency of the CSR clock is 150-250 MHz, and the MDC clock is CSR clock/102. 4’b0100: The frequency of the CSR clock is 250-300 MHz, and the MDC clock is CSR clock/124. 4’b0110 and 4’b0111: Reserved When Bit 5 is set, you can achieve higher frequency of the MDC clock than the frequency limit of 2.5 MHz (specified in the IEEE Std 802.3) and program a clock divider of lower value. For example, when CSR clock is of 100 MHz frequency and you program these bits as 4’b1010, then the resultant MDC clock is of 12.5 MHz which is outside the limit of IEEE 802.3 specified range. Program the following values only if the interfacing chips support faster MDC clocks: 4’b1000: CSR clock/4 4’b1001: CSR clock/6 4’b1010: CSR clock/8 4’b1011: CSR clock/10 4’b1100: CSR clock/12 4’b1101: CSR clock/14 4’b1110: CSR clock/16 4’b1111: CSR clock/18	R/W

Table 6.6 GMII_Address Register Contents (2/2)

Bit Position	Bit Name	Function	R/W
b1	GW	GMII Write When set, this bit indicates to the PHY that this is a Write operation using the GMII Data register. If this bit is not set, it indicates that this is a Read operation, that is, placing the data in the GMII Data register.	R/W
b0	GB	GMII Busy This bit should read logic 0 before writing to GMII Address Register (GMII_Address) and GMII Data Register (GMII_Data). During a PHY register access, the software sets this bit to 1'b1 to indicate that a Read or Write access is in progress. GMII Data Register (GMII_Data) is invalid until this bit is cleared by the GMAC. Therefore, GMII Data Register (GMII_Data) should be kept valid until the GMAC clears this bit during a PHY Write operation. Similarly, for a read operation, the contents of GMII Data Register (GMII_Data) are not valid until this bit is cleared. The subsequent read or write operation should happen only after the previous operation is complete. Because there is no acknowledgment from the PHY to GMAC after a read or write operation is completed, there is no change in the functionality of this bit even when the PHY is not present.	R/W

6.4.4 GMII_Data — GMII Data Register

Address: GMAC1: 4400 0014h
GMAC2: 4400 2014h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	GD															
Value after reset	0	0	0	0	0	0	0	X	X	X	X	X	X	X	X	X

Table 6.7 GMII_Data Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b16	Reserved	Reserved	R
b15 to b0	GD	GMII Data This field 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.	R/W

6.4.5 Flow_Control — Flow Control Register

Address: GMAC1: 4400 0018h
GMAC2: 4400 2018h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	PT															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	DZPQ	—	PLT		UP	RFE	TFE	FCA_BPA
Value after reset	X	X	X	X	X	X	X	X	0	X	0	0	0	0	0	0

Table 6.8 Flow_Control Register Contents (1/2)

Bit Position	Bit Name	Function	R/W
b31 to b16	PT	Pause Time This field holds the value to be used in the Pause Time field in the transmit control frame. Consecutive writes to this register should be performed only after at least four clock cycles in the destination clock domain.	R/W
b15 to b8	Reserved	Reserved	R
b7	DZPQ	Disable Zero-Quanta Pause When this bit is set, it disables the automatic generation of the Zero-Quanta Pause Control frames on the de-assertion of the flow-control signal from the FIFO layer. When this bit is reset, normal operation with automatic Zero-Quanta Pause Control frame generation is enabled.	R/W
b6	Reserved	Reserved	R
b5, b4	PLT	Pause Low Threshold This field configures the threshold of the PAUSE timer at which the input flow control signal is checked for automatic retransmission of PAUSE Frame. The threshold values should be always less than the Pause Time configured in Bits[31:16]. For example, if PT = 100h (256 slot-times), and PLT = 2'b01, then a second PAUSE frame is automatically transmitted if the flow control signal is asserted at 228 (256 - 28) slot times after the first PAUSE frame is transmitted. The following list provides the threshold values for different values: 2'b00: The threshold is Pause time minus 4 slot times (PT - 4 slot times). 2'b01: The threshold is Pause time minus 28 slot times (PT - 28 slot times). 2'b10: The threshold is Pause time minus 144 slot times (PT - 144 slot times). 2'b11: The threshold is Pause time minus 256 slot times (PT - 256 slot times). The slot time is defined as the time taken to transmit 512 bits (64 bytes) on the GMII or MII interface.	R/W
b3	UP	Unicast Pause Frame Detect A pause frame is processed when it has the unique multicast address specified in the IEEE Std 802.3. When this bit is set, the GMAC can also detect Pause frames with unicast address of the station. This unicast address should be as specified in the MAC Address0 High Register and MAC Address0 Low Register. When this bit is reset, the GMAC only detects Pause frames with unique multicast address. ***Note)*** The GMAC does not process a Pause frame if the multicast address of received frame is different from the unique multicast address.	R/W

Table 6.8 Flow_Control Register Contents (2/2)

Bit Position	Bit Name	Function	R/W
b2	RFE	Receive Flow Control Enable When this bit is set, the GMAC decodes the received Pause frame and disables its transmitter for a specified (Pause) time. When this bit is reset, the decode function of the Pause frame is disabled.	R/W
b1	TFE	Transmit Flow Control Enable In the full-duplex mode, when this bit is set, the GMAC enables the flow control operation to transmit Pause frames. When this bit is reset, the flow control operation in the GMAC is disabled, and the GMAC does 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 back-pressure feature is disabled.	R/W
b0	FCA_BPA	Flow Control Busy or Backpressure Activate This bit initiates a Pause Control frame in the full-duplex mode and activates the backpressure function in the half-duplex mode if the TFE bit is set. In the full-duplex mode, this bit should be read as 1’b0 before writing to the Flow Control register. To initiate a Pause control frame, the Application must set this bit to 1’b1. During a transfer of the Control Frame, this bit continues to be set to signify that a frame transmission is in progress. After the completion of Pause control frame transmission, the GMAC resets this bit to 1’b0. The Flow Control register should not be written to until this bit is cleared. In the half-duplex mode, when this bit is set (and TFE is set), then backpressure is asserted by the GMAC. During backpressure, when the GMAC receives a new frame, the transmitter starts sending a JAM pattern resulting in a collision. This control register bit is logically ORed with the flow control input signal for the backpressure function. When the GMAC is configured for the full-duplex mode, backpressure is automatically disabled.	R/W

6.4.6 VLAN_Tag — VLAN Tag Register

Address: GMAC1: 4400 001Ch
GMAC2: 4400 201Ch

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	VTHM	ESVL	VTIM	ETV
Value after reset	0	X	X	X	X	X	X	X	X	X	X	X	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	VL															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6.9 VLAN_Tag Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b20	Reserved	Reserved	R
b19	VTHM	VLAN Tag Hash Table Match Enable When set, the most significant four bits of the VLAN tag's CRC are used to index the content of VLAN Hash Table Register (VLAN_Hash_Table_Reg). A value of 1 in the VLAN Hash Table register, corresponding to the index, indicates that the frame matched the VLAN hash table. When Bit 16 (ETV) is set, the CRC of the 12-bit VLAN Identifier (VID) is used for comparison whereas when ETV is reset, the CRC of the 16-bit VLAN tag is used for comparison. When reset, the VLAN Hash Match operation is not performed.	R/W
b18	ESVL	Enable S-VLAN When this bit is set, the GMAC transmitter and receiver also consider the S-VLAN (Type = 0x88A8) frames as valid VLAN tagged frames.	R/W
b17	VTIM	VLAN Tag Inverse Match Enable When set, this bit enables the VLAN Tag inverse matching. The frames that do not have matching VLAN Tag are marked as matched. When reset, this bit enables the VLAN Tag perfect matching. The frames with matched VLAN Tag are marked as matched.	R/W
b16	ETV	Enable 12-Bit VLAN Tag Comparison When this bit is set, a 12-bit VLAN identifier is used for comparing and filtering instead of the complete 16-bit VLAN tag. Bits [11:0] of VLAN tag are compared with the corresponding field in the received VLAN-tagged frame. Similarly, when enabled, only 12 bits of the VLAN tag in the received frame are used for hash-based VLAN filtering. When this bit is reset, all 16 bits of the 15th and 16th bytes of the received VLAN frame are used for comparison and VLAN hash filtering.	R/W
b15 to b0	VL	VLAN Tag Identifier for Receive Frames This field contains the 802.1Q VLAN tag to identify the VLAN frames and is compared to the 15th and 16th bytes of the frames being received for VLAN frames. The following list describes the bits of this field: Bits [15:13]: User Priority Bit 12: Canonical Format Indicator (CFI) or Drop Eligibility Indicator (DEI) Bits[11:0]: VLAN tag's VLAN Identifier (VID) field When the ETV bit is set, only the VID (Bits[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 16th bytes for VLAN tag comparison, and declares all frames with a Type field value of 0x8100 or 0x88a8 as VLAN frames.	R/W

6.4.7 Version — Version Register

Address: GMAC1: 4400 0020h
GMAC2: 4400 2020h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	VER															
Value after reset	0	0	1	0	0	0	0	X	0	0	1	1	0	1	1	1

Table 6.10 Version Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b16	Reserved	Reserved	R
b15 to b0	VER	Version (GMAC1: 0x2037, GMAC2: 0x2137)	R

6.4.8 Debug — Debug Register

Address: GMAC1: 4400 0024h
GMAC2: 4400 2024h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	TXSTSFSTS	TXFSTS	—	TWCSTS	TRCSTS		TXPAUSED	TFCSTS		TPESTS
Value after reset	X	X	X	X	X	X	0	0	X	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	RXFSTS		—	RRCSTS		RWCSTS	—	RFCFCSTS		RPESTS
Value after reset	X	X	X	X	X	X	0	0	X	0	0	0	X	0	0	0

Table 6.11 Debug Register Contents (1/2)

Bit Position	Bit Name	Function	R/W
b31 to b26	Reserved	Reserved	R
b25	TXSTSFSTS	MTL TxStatus FIFO Full Status When high, this bit indicates that the MTL (MAC Transaction Layer) TxStatus FIFO is full. Therefore, the MTL cannot accept any more frames for transmission.	R
b24	TXFSTS	MTL Tx FIFO Not Empty Status When high, this bit indicates that the MTL Tx FIFO is not empty and some data is left for transmission.	R
b23	Reserved	Reserved	R
b22	TWCSTS	MTL Tx FIFO Write Controller Active Status When high, this bit indicates that the MTL Tx FIFO Write Controller is active and transferring data to the Tx FIFO.	R
b21, b20	TRCSTS	MTL Tx FIFO Read Controller Status This field indicates the state of the Tx FIFO Read Controller: 2'b00: IDLE state 2'b01: READ state (transferring data to GMAC transmitter) 2'b10: Waiting for TxStatus from GMAC transmitter 2'b11: Writing the received TxStatus or flushing the Tx FIFO	R
b19	TXPAUSED	GMAC transmitter in PAUSE When high, this bit indicates that the GMAC transmitter is in the PAUSE condition (in the full-duplex only mode) and hence does not schedule any frame for transmission.	R
b18, b17	TFCSTS	GMAC Transmit Frame Controller Status This field indicates the state of the GMAC Transmit Frame Controller module: 2'b00: IDLE state 2'b01: Waiting for Status of previous frame or IFG or backoff period to be over 2'b10: Generating and transmitting a PAUSE control frame (in the full-duplex mode) 2'b11: Transferring input frame for transmission	R
b16	TPESTS	GMAC GMII or MII Transmit Protocol Engine Status When high, this bit indicates that the GMAC GMII or MII transmit protocol engine is actively transmitting data and is not in the IDLE state.	R
b15 to b10	Reserved	Reserved	R
b9, b8	RXFSTS	MTL Rx FIFO Fill-level Status This field gives the status of the fill-level of the Rx FIFO: 2'b00: Rx FIFO Empty 2'b01: Rx FIFO fill level is below the flow-control deactivate threshold 2'b10: Rx FIFO fill level is above the flow-control activate threshold 2'b11: Rx FIFO Full	R

Table 6.11 Debug Register Contents (2/2)

Bit Position	Bit Name	Function	R/W
b7	Reserved	Reserved	R
b6, b5	RRCSTS	MTL Rx FIFO Read Controller State This field gives the state of the Rx FIFO read Controller: 2'b00: IDLE state 2'b01: Reading frame data 2'b10: Reading frame status (or timestamp) 2'b11: Flushing the frame data and status	R
b4	RWCSTS	MTL Rx FIFO Write Controller Active Status When high, this bit indicates that the MTL Rx FIFO Write Controller is active and is transferring a received frame to the FIFO.	R
b3	Reserved	Reserved	R
b2, b1	RFCFCSTS	GMAC Receive Frame Controller FIFO Status When high, this field indicates the active state of the small FIFO Read and Write controllers of the GMAC Receive Frame Controller Module.	R
b0	RPESTS	GMAC GMII or MII Receive Protocol Engine Status When high, this bit indicates that the GMAC GMII or MII receive protocol engine is actively receiving data and not in IDLE state.	R

6.4.9 Remote_Wake_Up_Frame_Filter — Remote Wake-Up Frame Filter Register

Address: GMAC1: 4400 0028h
GMAC2: 4400 2028h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	WKUPFRMFTR															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	WKUPFRMFTR															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6.12 Remote_Wake_Up_Frame_Filter Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	WKUPFRMFTR	Remote Wake-Up Frame Filter	R/W

6.4.10 PMT_Control_Status — PMT Control and Status Register

Address: GMAC1: 4400 002Ch
GMAC2: 4400 202Ch

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	RWKFILTRST	—	—	—	—	RWKPTR			—	—	—	—	—	—	—	—
Value after reset	0	X	X	X	X	0	0	0	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	GLBLUCAST	—	—	RWKPRCVD	MGKPRCVD	—	—	RWKPKTEN	MGKPKTEN	PWRDWN
Value after reset	X	X	X	X	X	0	0	X	X	0	0	X	X	0	0	0

Table 6.13 PMT_Control_Status Register Contents (1/2)

Bit Position	Bit Name	Function	R/W
b31	RWKFILTRST	Wake-Up Frame Filter Register Pointer Reset When this bit is set, it resets the remote wake-up frame filter register pointer to 3'b000. It is automatically cleared after 1 clock cycle.	R/W
b30 to b27	Reserved	Reserved	R
b26 to b24	RWKPTR	Remote Wake-up FIFO Pointer This gives the current value (0 to 7) of the Remote Wake-up Frame filter register pointer. The contents of the Remote Wake-up Frame Filter Register are transferred to the Receive Clock domain when a write occurs to that register when this pointer value equals 7.	R
b23 to b11	Reserved	Reserved	R
b10	Reserved	Keep Initial value	R/W
b9	GLBLUCAST	Global Unicast When set, enables any unicast packet filtered by the MAC address recognition (DA filter) to be a wake-up frame.	R/W
b8, b7	Reserved	Reserved	R
b6	RWKPRCVD	Wake-Up Frame Received When set, this bit indicates the power management event is generated because of the reception of a wake-up frame. This bit is cleared by a Read into this register.	R
b5	MGKPRCVD	Magic Packet Received When set, this bit indicates that the power management event is generated because of the reception of a magic packet. This bit is cleared by a Read into this register.	R
b4, b3	Reserved	Reserved	R
b2	RWKPKTEN	Wake-Up Frame Enable When set, enables generation of a power management event because of wake-up frame reception.	R/W
b1	MGKPKTEN	Magic Packet Enable When set, enables generation of a power management event because of magic packet reception.	R/W

Table 6.13 PMT_Control_Status Register Contents (2/2)

Bit Position	Bit Name	Function	R/W
b0	PWRDWN	Power Down When set, the GMAC receiver drops all received frames until it receives the expected magic packet or wake-up frame. This bit is then self-cleared and the power-down mode is disabled. The Software can also clear this bit before the expected magic packet or wake-up frame is received. The frames, received by the GMAC after this bit is cleared, are forwarded to the application. This bit must only be set when the Magic Packet Enable, Global Unicast, or Wake-Up Frame Enable bit is set high. ***Note)*** You can gate-off the CSR clock (GMAC[m]_HCLK) during the power-down mode. However, when the CSR clock is gated-off, you cannot perform any read or write operations on this register. Therefore, the Software cannot clear this bit.	R/W

6.4.11 LPI_Control_Status — LPI Control and Status Register

Address: GMAC1: 4400 0030h
GMAC2: 4400 2030h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	LPITXA	—	PLS	LPIEN
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	0	X	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	RLPIST	TLPIST	—	—	—	—	RLPIEX	RLPIEN	TLPIEX	TLPIEN
Value after reset	X	X	X	X	X	X	0	0	X	X	X	X	0	0	0	0

Table 6.14 LPI_Control_Status Register Contents (1/2)

Bit Position	Bit Name	Function	R/W
b31 to b20	Reserved	Reserved	R
b19	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 the LPITXA and LPIEN bits are set to 1, the GMAC enters the LPI mode only after all outstanding frames (in the core) and pending frames (in the application interface) have been transmitted. The GMAC comes out of the LPI mode when the application sends any frame 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. If TX FIFO Flush (Bit 20) of Operation Mode Register (Operation_Mode) is set when the GMAC is in the LPI mode, the GMAC exits the LPI mode. 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.	R/W
b18	Reserved	Reserved	R
b17	PLS	PHY Link Status This bit indicates 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 set, the link is considered to be okay (up) and when reset, the link is considered to be down.	R/W
b16	LPIEN	LPI Enable When set, this bit instructs the GMAC Transmitter to enter the LPI state. When reset, this bit instructs the GMAC to exit the LPI state and resume normal transmission. This bit is cleared when the LPITXA bit is set and the GMAC exits the LPI state because of the arrival of a new packet for transmission.	R/W
b15 to b10	Reserved	Reserved	R
b9	RLPIST	Receive LPI State When set, this bit indicates that the GMAC is receiving the LPI pattern on the GMII or MII interface.	R
b8	TLPIST	Transmit LPI State When set, this bit indicates that the GMAC is transmitting the LPI pattern on the GMII or MII interface.	R
b7 to b4	Reserved	Reserved	R
b3	RLPIEX	Receive LPI Exit When set, this bit indicates that the GMAC Receiver has stopped receiving the LPI pattern on the GMII or MII interface, exited the LPI state, and resumed the normal reception. This bit is cleared by a read into this register. ***Note)*** This bit may not get set if the GMAC stops receiving the LPI pattern for a very short duration, such as, less than 3 clock cycles of CSR clock.	R

Table 6.14 LPI_Control_Status Register Contents (2/2)

Bit Position	Bit Name	Function	R/W
b2	RLPIEN	Receive LPI Entry When 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*)** This bit may not get set if the GMAC stops receiving the LPI pattern for a very short duration, such as, less than 3 clock cycles of CSR clock.	R
b1	TLPIEX	Transmit LPI Exit When 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.	R
b0	TLPIEN	Transmit LPI Entry When 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.	R

6.4.12 LPI_Timers_Control — LPI Timers Control Register

Address: GMAC1: 4400 0034h
GMAC2: 4400 2034h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	LST									
Value after reset	X	X	X	X	X	X	1	1	1	1	1	0	1	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TWT															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6.15 LPI_Timers_Control Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b26	Reserved	Reserved	R
b25 to b16	LST	LPI LS Timer This field specifies 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 the 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.	R/W
b15 to b0	TWT	LPI TW Timer This field specifies the minimum time (in microseconds) for which the GMAC waits after it stops 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.	R/W

6.4.13 Interrupt_Status — Interrupt Status Register

Address: GMAC1: 4400 0038h
GMAC2: 4400 2038h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	LPIIS	TSIS	—	MMCRXIPIS	MMCTXIS	MMCRXIS	MMCIS	PMTIS	—	—	—
Value after reset	X	X	X	X	X	0	0	X	0	0	0	0	0	X	X	X

Table 6.16 Interrupt_Status Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b11	Reserved	Reserved	R
b10	LPIIS	LPI Interrupt Status This bit is set for any LPI state entry or exit in the GMAC Transmitter or Receiver. This bit is cleared on reading Bit 0 of LPI Control and Status Register (LPI_Control_Status).	R
b9	TSIS	Timestamp Interrupt Status This bit is set when any of the following conditions is true: • 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. • The Auxiliary snapshot trigger is asserted. This bit is cleared on reading Bit 0 of the Timestamp Status Register (Timestamp_Status). If default Timestamping is enabled, when set, this bit indicates that the system time value is equal to or exceeds the value specified in the Target Time registers. In this mode, this bit is cleared after the completion of the read of this bit.	R
b8	Reserved	Reserved	R
b7	MMCRXIPIS	MMC Receive Checksum Offload Interrupt Status This bit is set high when an interrupt is generated in the MMC Receive Checksum Offload Interrupt Register. This bit is cleared when all the bits in this interrupt register are cleared.	R
b6	MMCTXIS	MMC Transmit Interrupt Status This bit is set high when an interrupt is generated in the MMC Transmit Interrupt Register. This bit is cleared when all the bits in this interrupt register are cleared.	R
b5	MMCRXIS	MMC Receive Interrupt Status This bit is set high when an interrupt is generated in the MMC Receive Interrupt Register. This bit is cleared when all the bits in this interrupt register are cleared.	R
b4	MMCIS	MMC Interrupt Status This bit is set high when any of the Bits [7:5] is set high and cleared only when all of these bits are low.	R
b3	PMTIS	PMT Interrupt Status This bit is set when a Magic packet or Wake-on-LAN frame is received in the power-down mode (see Bits 5 and 6 in the PMT Control and Status Register). This bit is cleared when both Bits[6:5] are cleared because of a read operation to the PMT Control and Status register.	R
b2 to b0	Reserved	Reserved	R

6.4.14 Interrupt_Mask — Interrupt Mask Register

Address: GMAC1: 4400 003Ch
GMAC2: 4400 203Ch

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	LPIIM	TSIM	—	—	—	—	—	PMTIM	—	—	—
Value after reset	X	X	X	X	X	0	0	X	X	X	X	X	0	X	X	X

Table 6.17 Interrupt_Mask Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b11	Reserved	Reserved	R
b10	LPIIM	LPI Interrupt Mask When set, this bit disables the assertion of the interrupt signal because of the setting of the LPI Interrupt Status bit in Interrupt Status Register (Interrupt_Status).	R/W
b9	TSIM	Timestamp Interrupt Mask When set, this bit disables the assertion of the interrupt signal because of the setting of Timestamp Interrupt Status bit in Interrupt Status Register (Interrupt_Status).	R/W
b8 to b4	Reserved	Reserved	R
b3	PMTIM	PMT Interrupt Mask When set, this bit disables the assertion of the interrupt signal because of the setting of PMT Interrupt Status bit in Interrupt Status Register (Interrupt_Status).	R/W
b2 to b0	Reserved	Reserved	R

6.4.15 MAC_Address0_High — MAC Address 0 High Register

Address: GMAC1: 4400 0040h
GMAC2: 4400 2040h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	AE	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ADDRHI															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Table 6.18 MAC_Address0_High Register Contents

Bit Position	Bit Name	Function	R/W
b31	AE	Address Enable This bit is always set to 1.	R
b30 to b16	Reserved	Reserved	R
b15 to b0	ADDRHI	MAC Address0 [47:32] This field contains the upper 16 bits (47:32) of the first 6-byte MAC address. The GMAC uses this field for filtering the received frames and inserting the MAC address in the Transmit Flow Control (Pause) Frames.	R/W

6.4.16 MAC_Address0_Low — MAC Address 0 Low Register

Address: GMAC1: 4400 0044h
GMAC2: 4400 2044h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	ADDRLO															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ADDRLO															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Table 6.19 MAC_Address0_Low Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	ADDRLO	MAC Address0 [31:0] This field contains the lower 32 bits of the first 6-byte MAC address. This is used by the GMAC for filtering the received frames and inserting the MAC address in the Transmit Flow Control (Pause) Frames.	R/W

6.4.17 MAC_Address[n]_High — MAC Address [n] High Register (n = 1..17)

Address: GMAC1: 4400 0040h + 8h × n (n = 1..15), 4400 0800h (n = 16), 4400 0808h (n = 17)
GMAC2: 4400 2040h + 8h × n (n = 1..15), 4400 2800h (n = 16), 4400 2808h (n = 17)

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	AE	SA	MBC						—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ADDRHI															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Table 6.20 MAC_Address[n]_High Register Contents

Bit Position	Bit Name	Function	R/W
b31	AE	Address Enable When this bit is set, the address filter module uses the [n]th MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering.	R/W
b30	SA	Source Address When this bit is set, the MAC Address[n][47:0] is used to compare with the SA fields of the received frame. When this bit is reset, the MAC Address[n][47:0] is used to compare with the DA fields of the received frame.	R/W
b29 to b24	MBC	Mask Byte Control These bits are mask control bits for comparison of each of the MAC Address bytes. When set high, the GMAC does not compare the corresponding byte of received DA or SA with the contents of MAC Address1 registers. Each bit controls the masking of the bytes as follows: Bit 29: MAC_Address[n]_High [15:8] Bit 28: MAC_Address[n]_High [7:0] Bit 27: MAC_Address[n]_Low [31:24] ... Bit 24: MAC_Address[n]_Low [7:0] You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.	R/W
b23 to b16	Reserved	Reserved	R
b15 to b0	ADDRHI	MAC Address[n] [47:32] This field contains the upper 16 bits (47:32) of the [n]th 6-byte MAC address.	R/W

6.4.18 MAC_Address[n]_Low — MAC Address [n] Low Register (n = 1..17)

Address: GMAC1: 4400 0044h +8h × n (n = 1..15), 4400 0804h (n = 16), 4400 080Ch (n = 17)
GMAC2: 4400 2044h +8h × n (n = 1..15), 4400 2804h (n = 16), 4400 280Ch (n = 17)

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	ADDRLO															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ADDRLO															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Table 6.21 MAC_Address[n]_Low Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	ADDRLO	MAC Address [n] [31:0] This field contains the lower 32 bits of the [n]th 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.	R/W

6.4.19 WDog_Timeout — Watchdog Timeout Register

Address: GMAC1: 4400 00DCh
GMAC2: 4400 20DCh

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PWE
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	WTO													
Value after reset	X	X	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6.22 WDog_Timeout Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b17	Reserved	Reserved	R
b16	PWE	Programmable Watchdog Enable When this bit is set and Bit 23 (WD) of MAC Configuration Register (MAC_Configuration) is reset, the WTO field (Bits[13:0]) is used as watchdog timeout for a received frame. When this bit is cleared, the watchdog timeout for a received frame is controlled by the setting of Bit 23 (WD) and Bit 20 (JE) in MAC Configuration Register (MAC_Configuration).	R/W
b15, b14	Reserved	Reserved	R
b13 to b0	WTO	Watchdog Timeout When Bit 16 (PWE) is set and Bit 23 (WD) of MAC Configuration Register (MAC_Configuration) is reset, this field is used as watchdog timeout for a received frame. If the length of a received frame exceeds the value of this field, such frame is terminated and declared as an error frame. ***Note)*** When Bit 16 (PWE) is set, the value in this field should be more than 1,522 (0x05F2). Otherwise, the IEEE Std 802.3-specified valid tagged frames are declared as error frames and are dropped.	R/W

6.4.20 MMC_Control — MMC Control Register

Address: GMAC1: 4400 0100h
GMAC2: 4400 2100h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	UCDBC	—	—	CNTPRSTLVL	CNTPRST	CNTFREEZ	RSTONRD	CNTSTOPRO	CNTRST
Value after reset	X	X	X	X	X	X	X	0	X	X	0	0	0	0	0	0

Table 6.23 MMC_Control Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b9	Reserved	Reserved	R
b8	UCDBC	Update MMC Counters for Dropped Broadcast Frames When set, this bit enables GMAC to update all the related MMC Counters for Broadcast frames dropped due to setting of DBF bit (Disable Broadcast Frames) of MAC Frame Filter Register at offset 0x0004. When reset, MMC Counters are not updated for dropped Broadcast frames.	R/W
b7, b6	Reserved	Reserved	R
b5	CNTPRSTLVL	Full-Half Preset When low and bit 4 is set, all MMC counters get preset to almost-half value. All octet counters get preset to 0x7FFF_F800 (half – 2 KBytes) and all frame-counters gets preset to 0x7FFF_FFF0 (half - 16). When this bit is high and bit 4 is set, all MMC counters get preset to almost-full value. All octet counters get preset to 0xFFFF_F800 (full – 2 KBytes) and all frame-counters gets preset to 0xFFFF_FFF0 (full - 16). For 16-bit counters, the almost-half preset values are 0x7800 and 0x7FF0 for the respective octet and frame counters. Similarly, the almost-full preset values for the 16-bit counters are 0xF800 and 0xFFF0.	R/W
b4	CNTPRST	Counters Preset When this bit is set, all counters are initialized or preset to almost full or almost half according to bit 5. This bit is cleared automatically after 1 clock cycle. This bit, along with bit 5, is useful for debugging and testing the assertion of interrupts because of MMC counter becoming half-full or full.	R/W
b3	CNTFREEZ	MMC Counter Freeze When this bit is set, it freezes all MMC counters to their current value. Until this bit is reset to 0, no MMC counter is updated because of any transmitted or received frame. If any MMC counter is read with the Reset on Read bit set, then that counter is also cleared in this mode.	R/W
b2	RSTONRD	Reset on Read When this bit is set, the MMC counters are reset to zero after Read (self-clearing after reset). The counters are cleared when the least significant byte lane (bits[7:0]) is read.	R/W
b1	CNTSTOPRO	Counters Stop Rollover When this bit is set, after reaching maximum value, the counter does not roll over to zero.	R/W
b0	CNTRST	Counters Reset When this bit is set, all counters are reset. This bit is cleared automatically after one clock cycle.	R/W

6.4.21 MMC_Receive_Interrupt — MMC Receive Interrupt Register

Address: GMAC1: 4400 0104h
GMAC2: 4400 2104h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	RXCTRLFIS	RXRCVERRFIS	RXWDOGFIS	RXVLANGBFIS	RXFOVFIS	RXPAUSFIS	RXORANGEFIS	RXLENERFIS	RXUCGFIS	RX1024TMAXOCTGBFIS
Value after reset	X	X	X	X	X	X	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RX512T1023OCTGBFIS	RX256T511OCTGBFIS	RX128T255OCTGBFIS	RX65T127OCTGBFIS	RX64OCTGBFIS	RXOSIZEGFIS	RXUSIZEGFIS	RXJABERFIS	RXRUNTFIS	RXALGNERFIS	RXCRCERFIS	RXMCGFIS	RXBCGFIS	RXGOCTIS	RXGBOCTIS	RXGBFRMIS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6.24 MMC_Receive_Interrupt Register Contents (1/2)

Bit Position	Bit Name	Function	R/W
b31 to b26	Reserved	Reserved	R
b25	RXCTRLFIS	MMC Receive Control Frame Counter Interrupt Status This bit is set when the Rx_Control_Frames_Good counter reaches half of the maximum value or the maximum value.	R
b24	RXRCVERRFIS	MMC Receive Error Frame Counter Interrupt Status This bit is set when the Rx_Receive_Error_Frames counter reaches half of the maximum value or the maximum value.	R
b23	RXWDOGFIS	MMC Receive Watchdog Error Frame Counter Interrupt Status This bit is set when the Rx_Watchdog_Error_Frames counter reaches half of the maximum value or the maximum value.	R
b22	RXVLANGBFIS	MMC Receive VLAN Good Bad Frame Counter Interrupt Status This bit is set when the Rx_VLAN_Frames_Good_Bad counter reaches half of the maximum value or the maximum value.	R
b21	RXFOVFIS	MMC Receive FIFO Overflow Frame Counter Interrupt Status This bit is set when the Rx_FIFO_Overflow_Frames counter reaches half of the maximum value or the maximum value.	R
b20	RXPAUSFIS	MMC Receive Pause Frame Counter Interrupt Status This bit is set when the Rx_Pause_Frames counter reaches half of the maximum value or the maximum value.	R
b19	RXORANGEFIS	MMC Receive Out Of Range Error Frame Counter Interrupt Status This bit is set when the Rx_Out_Of_Range_Type_Frames counter reaches half of the maximum value or the maximum value.	R
b18	RXLENERFIS	MMC Receive Length Error Frame Counter Interrupt Status This bit is set when the Rx_Length_Error_Frames counter reaches half of the maximum value or the maximum value.	R
b17	RXUCGFIS	MMC Receive Unicast Good Frame Counter Interrupt Status This bit is set when the Rx_Unicast_Frames_Good counter reaches half of the maximum value or the maximum value.	R
b16	RX1024TMAXOCTGBFIS	MMC Receive 1024 to Maximum Octet Good Bad Frame Counter Interrupt Status This bit is set when the Rx_1024ToMaxOctets_Frames_Good_Bad counter reaches half of the maximum value or the maximum value.	R
b15	RX512T1023OCTGBFIS	MMC Receive 512 to 1023 Octet Good Bad Frame Counter Interrupt Status This bit is set when the Rx_512To1023Octets_Frames_Good_Bad counter reaches half of the maximum value or the maximum value.	R
b14	RX256T511OCTGBFIS	MMC Receive 256 to 511 Octet Good Bad Frame Counter Interrupt Status This bit is set when the Rx_256To511Octets_Frames_Good_Bad counter reaches half of the maximum value or the maximum value.	R

Table 6.24 MMC_Receive_Interrupt Register Contents (2/2)

Bit Position	Bit Name	Function	R/W
b13	RX128T255OCTGBFIS	MMC Receive 128 to 255 Octet Good Bad Frame Counter Interrupt Status This bit is set when the Rx_128To255Octets_Frames_Good_Bad counter reaches half of the maximum value or the maximum value.	R
b12	RX65T127OCTGBFIS	MMC Receive 65 to 127 Octet Good Bad Frame Counter Interrupt Status This is set when the Rx_65To127Octets_Frames_Good_Bad counter reaches half of the maximum value or the maximum value.	R
b11	RX64OCTGBFIS	MMC Receive 64 Octet Good Bad Frame Counter Interrupt Status This bit is set when the Rx_64Octets_Frames_Good_Bad counter reaches half of the maximum value or the maximum value.	R
b10	RXOSIZEGFIS	MMC Receive Oversize Good Frame Counter Interrupt Status This bit is set when the Rx_Oversize_Frames_Good counter reaches half of the maximum value or the maximum value.	R
b9	RXUSIZEGFIS	MMC Receive Undersize Good Frame Counter Interrupt Status This bit is set when the Rx_Undersize_Frames_Good counter reaches half of the maximum value or the maximum value.	R
b8	RXJABERFIS	MMC Receive Jabber Error Frame Counter Interrupt Status This bit is set when the Rx_Jabber_Error_Frames counter reaches half of the maximum value or the maximum value.	R
b7	RXRUNTFIS	MMC Receive Runt Frame Counter Interrupt Status This bit is set when the Rx_Runt_Error_Frames counter reaches half of the maximum value or the maximum value.	R
b6	RXALGNERFIS	MMC Receive Alignment Error Frame Counter Interrupt Status This bit is set when the Rx_Alignment_Error_Frames counter reaches half of the maximum value or the maximum value.	R
b5	RXCRCERFIS	MMC Receive CRC Error Frame Counter Interrupt Status This bit is set when the Rx_CRC_Error_Frames counter reaches half of the maximum value or the maximum value.	R
b4	RXMCGFIS	MMC Receive Multicast Good Frame Counter Interrupt Status This bit is set when the Rx_Multicast_Frames_Good counter reaches half of the maximum value or the maximum value.	R
b3	RXBCGFIS	MMC Receive Broadcast Good Frame Counter Interrupt Status. This bit is set when the Rx_Broadcast_Frames_Good counter reaches half of the maximum value or the maximum value.	R
b2	RXGOCTIS	MMC Receive Good Octet Counter Interrupt Status. This bit is set when the Rx_Octet_Count_Good counter reaches half of the maximum value or the maximum value.	R
b1	RXGBOCTIS	MMC Receive Good Bad Octet Counter Interrupt Status This bit is set when the Rx_Octet_Count_Good_Bad counter reaches half of the maximum value or the maximum value.	R
b0	RXGBFRMIS	MMC Receive Good Bad Frame Counter Interrupt Status This bit is set when the Rx_Frames_Count_Good_Bad counter reaches half of the maximum value or the maximum value.	R

6.4.22 MMC_Transmit_Interrupt — MMC Transmit Interrupt Register

Address: GMAC1: 4400 0108h
GMAC2: 4400 2108h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	TXOSIZEGFIS	TXVLANGFIS	TXPAUSFIS	TXEXDEFFIS	TXGFRMIS	TXGOCTIS	TXCARERFIS	TXEXCOLFIS	TXLATCOLFIS	TXDEFFIS
Value after reset	X	X	X	X	X	X	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TXMCOLGFIS	TXSCOLGFIS	TXUFLOWERFIS	TXBCGBFIS	TXMCGBFIS	TXUCGBFIS	TX1024TMAXOCTGBFIS	TX512T1023OCTGBFIS	TX256T511OCTGBFIS	TX128T255OCTGBFIS	TX65T127OCTGBFIS	TX64OCTGBFIS	TXMCGFIS	TXBCGFIS	TXGBFRMIS	TXGBOCTIS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6.25 MMC_Transmit_Interrupt Register Contents (1/2)

Bit Position	Bit Name	Function	R/W
b31 to b26	Reserved	Reserved	R
b25	TXOSIZEGFIS	MMC Transmit Oversize Good Frame Counter Interrupt Status This bit is set when the Tx_OSize_Frames_Good counter reaches half of the maximum value or the maximum value.	R
b24	TXVLANGFIS	MMC Transmit VLAN Good Frame Counter Interrupt Status This bit is set when the Tx_VLAN_Frames_Good counter reaches half of the maximum value or the maximum value.	R
b23	TXPAUSFIS	MMC Transmit Pause Frame Counter Interrupt Status This bit is set when the Tx_Pause_Frames counter reaches half of the maximum value or the maximum value.	R
b22	TXEXDEFFIS	MMC Transmit Excessive Deferral Frame Counter Interrupt Status This bit is set when the Tx_Excessive_Deferral_Error counter reaches half of the maximum value or the maximum value.	R
b21	TXGFRMIS	MMC Transmit Good Frame Counter Interrupt Status This bit is set when the Tx_Frame_Count_Good counter reaches half of the maximum value or the maximum value.	R
b20	TXGOCTIS	MMC Transmit Good Octet Counter Interrupt Status This bit is set when the Tx_Octet_Count_Good counter reaches half of the maximum value or the maximum value.	R
b19	TXCARERFIS	MMC Transmit Carrier Error Frame Counter Interrupt Status This bit is set when the Tx_Carrier_Error_Frames counter reaches half of the maximum value or the maximum value.	R
b18	TXEXCOLFIS	MMC Transmit Excessive Collision Frame Counter Interrupt Status This bit is set when the Tx_Excessive_Collision_Frames counter reaches half of the maximum value or the maximum value.	R
b17	TXLATCOLFIS	MMC Transmit Late Collision Frame Counter Interrupt Status This bit is set when the Tx_Late_Collision_Frames counter reaches half of the maximum value or the maximum value.	R
b16	TXDEFFIS	MMC Transmit Deferred Frame Counter Interrupt Status This bit is set when the Tx_Deferred_Frames counter reaches half of the maximum value or the maximum value.	R
b15	TXMCOLGFIS	MMC Transmit Multiple Collision Good Frame Counter Interrupt Status This bit is set when the Tx_Multiple_Collision_Good_Frames counter reaches half of the maximum value or the maximum value.	R
b14	TXSCOLGFIS	MMC Transmit Single Collision Good Frame Counter Interrupt Status This bit is set when the Tx_Single_Collision_Good_Frames counter reaches half of the maximum value or the maximum value.	R

Table 6.25 MMC_Transmit_Interrupt Register Contents (2/2)

Bit Position	Bit Name	Function	R/W
b13	TXUFLOWERFIS	MMC Transmit Underflow Error Frame Counter Interrupt Status This bit is set when the Tx_Underflow_Error_Frames counter reaches half of the maximum value or the maximum value.	R
b12	TXBCGBFIS	MMC Transmit Broadcast Good Bad Frame Counter Interrupt Status This bit is set when the Tx_Broadcast_Frames_Good_Bad counter reaches half of the maximum value or the maximum value.	R
b11	TXMCGBFIS	MMC Transmit Multicast Good Bad Frame Counter Interrupt Status This bit is set when the Tx_Multicast_Frames_Good_Bad counter reaches half of the maximum value or the maximum value.	R
b10	TXUCGBFIS	MMC Transmit Unicast Good Bad Frame Counter Interrupt Status This bit is set when the Tx_Unicast_Frames_Good_Bad counter reaches half of the maximum value or the maximum value.	R
b9	TX1024TMAXOCTGBFIS	MMC Transmit 1024 to Maximum Octet Good Bad Frame Counter Interrupt Status This bit is set when the Tx_1024ToMaxOctets_Frames_Good_Bad counter reaches half of the maximum value or the maximum value.	R
b8	TX512T1023OCTGBFIS	MMC Transmit 512 to 1023 Octet Good Bad Frame Counter Interrupt Status This bit is set when the Tx_512To1023Octets_Frames_Good_Bad counter reaches half of the maximum value or the maximum value.	R
b7	TX256T511OCTGBFIS	MMC Transmit 256 to 511 Octet Good Bad Frame Counter Interrupt Status This bit is set when the Tx_256To511Octets_Frames_Good_Bad counter reaches half of the maximum value or the maximum value.	R
b6	TX128T255OCTGBFIS	MMC Transmit 128 to 255 Octet Good Bad Frame Counter Interrupt Status This bit is set when the Tx_128To255Octets_Frames_Good_Bad counter reaches half of the maximum value or the maximum value.	R
b5	TX65T127OCTGBFIS	MMC Transmit 65 to 127 Octet Good Bad Frame Counter Interrupt Status This bit is set when the Tx_65To127Octets_Frames_Good_Bad counter reaches half of the maximum value or the maximum value.	R
b4	TX64OCTGBFIS	MMC Transmit 64 Octet Good Bad Frame Counter Interrupt Status. This bit is set when the Tx_64Octets_Frames_Good_Bad counter reaches half of the maximum value or the maximum value.	R
b3	TXMCGFIS	MMC Transmit Multicast Good Frame Counter Interrupt Status This bit is set when the Tx_Multicast_Frames_Good counter reaches half of the maximum value or the maximum value.	R
b2	TXBCGFIS	MMC Transmit Broadcast Good Frame Counter Interrupt Status This bit is set when the Tx_Broadcast_Frames_Good counter reaches half of the maximum value or the maximum value.	R
b1	TXGBFRMIS	MMC Transmit Good Bad Frame Counter Interrupt Status This bit is set when the Tx_Frame_Count_Good_Bad counter reaches half of the maximum value or the maximum value.	R
b0	TXGBOCTIS	MMC Transmit Good Bad Octet Counter Interrupt Status This bit is set when the Tx_Octet_Count_Good_Bad counter reaches half of the maximum value or the maximum value.	R

6.4.23 MMC_Receive_Interrupt_Mask — MMC Receive Interrupt Mask Register

Address: GMAC1: 4400 010Ch
GMAC2: 4400 210Ch

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	RXCTRLFIM	RXRCVERRFIM	RXWDOGFIM	RXVLANGBFIM	RXFOVFIM	RXPAUSFIM	RXORANGEFIM	RXLENERFIM	RXUCGFIM	RX1024TMAXOCTGBFIM
Value after reset	X	X	X	X	X	X	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RX512T1023OCTGBFIM	RX256T511OCTGBFIM	RX128T255OCTGBFIM	RX65T127OCTGBFIM	RX64OCTGBFIM	RXOSIZEGFIM	RXUSIZEGFIM	RXJABERFIM	RXRUNTFIM	RXALGNERFIM	RXCRCERFIM	RXMCGFIM	RXBCGFIM	RXGOCTIM	RXGBOCTIM	RXGBFRMIM
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6.26 MMC_Receive_Interrupt_Mask Register Contents (1/2)

Bit Position	Bit Name	Function	R/W
b31 to b26	Reserved	Reserved	R/W
b25	RXCTRLFIM	MMC Receive Control Frame Counter Interrupt Mask Setting this bit masks the interrupt generated when the Rx_Control_Frames_Good counter reaches half the maximum value, and also when it reaches the maximum value.	R/W
b24	RXRCVERRFIM	MMC Receive Error Frame Counter Interrupt Mask Setting this bit masks the interrupt generated when the Rx_Receive_Error_Frames error counter reaches half the maximum value, and also when it reaches the maximum value.	R/W
b23	RXWDOGFIM	MMC Receive Watchdog Error Frame Counter Interrupt Mask Setting this bit masks the interrupt generated when the Rx_Watchdog_Error_Frames counter reaches half of the maximum value or the maximum value.	R/W
b22	RXVLANGBFIM	MMC Receive VLAN Good Bad Frame Counter Interrupt Mask Setting this bit masks the interrupt generated when the Rx_VLAN_Frames_Good_Bad counter reaches half of the maximum value or the maximum value.	R/W
b21	RXFOVFIM	MMC Receive FIFO Overflow Frame Counter Interrupt Mask Setting this bit masks the interrupt generated when the Rx_FIFO_Overflow_Frames counter reaches half of the maximum value or the maximum value.	R/W
b20	RXPAUSFIM	MMC Receive Pause Frame Counter Interrupt Mask Setting this bit masks the interrupt generated when the Rx_Pause_Frames counter reaches half of the maximum value or the maximum value.	R/W
b19	RXORANGEFIM	MMC Receive Out Of Range Error Frame Counter Interrupt Mask Setting this bit masks the interrupt generated when the Rx_Out_Of_Range_Type_Frames counter reaches half of the maximum value or the maximum value.	R/W
b18	RXLENERFIM	MMC Receive Length Error Frame Counter Interrupt Mask Setting this bit masks the interrupt generated when the Rx_Length_Error_Frames counter reaches half of the maximum value or the maximum value.	R/W
b17	RXUCGFIM	MMC Receive Unicast Good Frame Counter Interrupt Mask Setting this bit masks the interrupt generated when the Rx_Unicast_Frames_Good counter reaches half of the maximum value or the maximum value.	R/W
b16	RX1024TMAXOCTGBFIM	MMC Receive 1024 to Maximum Octet Good Bad Frame Counter Interrupt Mask Setting this bit masks the interrupt generated when the Rx_1024ToMaxOctets_Frames_Good_Bad counter reaches half of the maximum value or the maximum value.	R/W
b15	RX512T1023OCTGBFIM	MMC Receive 512 to 1023 Octet Good Bad Frame Counter Interrupt Mask Setting this bit masks the interrupt generated when the Rx_512To1023Octets_Frames_Good_Bad counter reaches half of the maximum value or the maximum value.	R/W

Table 6.26 MMC_Receive_Interrupt_Mask Register Contents (2/2)

Bit Position	Bit Name	Function	R/W
b14	RX256T511OCTGBFIM	MMC Receive 256 to 511 Octet Good Bad Frame Counter Interrupt Mask Setting this bit masks the interrupt generated when the Rx_256To511Octets_Frames_Good_Bad counter reaches half of the maximum value or the maximum value.	R/W
b13	RX128T255OCTGBFIM	MMC Receive 128 to 255 Octet Good Bad Frame Counter Interrupt Mask Setting this bit masks the interrupt generated when the Rx_128To255Octets_Frames_Good_Bad counter reaches half of the maximum value or the maximum value.	R/W
b12	RX65T127OCTGBFIM	MMC Receive 65 to 127 Octet Good Bad Frame Counter Interrupt Mask Setting this bit masks the interrupt generated when the Rx_65To127Octets_Frames_Good_Bad counter reaches half of the maximum value or the maximum value.	R/W
b11	RX64OCTGBFIM	MMC Receive 64 Octet Good Bad Frame Counter Interrupt Mask Setting this bit masks the interrupt generated when the Rx_64Octets_Frames_Good_Bad counter reaches half of the maximum value or the maximum value.	R/W
b10	RXOSIZEGFIM	MMC Receive Oversize Good Frame Counter Interrupt Mask Setting this bit masks the interrupt generated when the Rx_Oversize_Frames_Good counter reaches half of the maximum value or the maximum value.	R/W
b9	RXUSIZEGFIM	MMC Receive Undersize Good Frame Counter Interrupt Mask Setting this bit masks the interrupt generated when the Rx_Undersize_Frames_Good counter reaches half of the maximum value or the maximum value.	R/W
b8	RXJABERFIM	MMC Receive Jabber Error Frame Counter Interrupt Mask Setting this bit masks the interrupt generated when the Rx_Jabber_Error_Frames counter reaches half of the maximum value or the maximum value.	R/W
b7	RXRUNTFIM	MMC Receive Runt Frame Counter Interrupt Mask Setting this bit masks the interrupt generated when the Rx_Runt_Error_Frames counter reaches half of the maximum value or the maximum value.	R/W
b6	RXALGNERFIM	MMC Receive Alignment Error Frame Counter Interrupt Mask Setting this bit masks the interrupt generated when the Rx_Alignment_Error_Frames counter reaches half of the maximum value or the maximum value.	R/W
b5	RXCRCERFIM	MMC Receive CRC Error Frame Counter Interrupt Mask Setting this bit masks the interrupt generated when the Rx_CRC_Error_Frames counter reaches half of the maximum value or the maximum value.	R/W
b4	RXMCGFIM	MMC Receive Multicast Good Frame Counter Interrupt Mask Setting this bit masks the interrupt generated when the Rx_Multicast_Frames_Good counter reaches half of the maximum value or the maximum value.	R/W
b3	RXBCGFIM	MMC Receive Broadcast Good Frame Counter Interrupt Mask Setting this bit masks the interrupt generated when the Rx_Broadcast_Frames_Good counter reaches half of the maximum value or the maximum value.	R/W
b2	RXGOCTIM	MMC Receive Good Octet Counter Interrupt Mask Setting this bit masks the interrupt generated when the Rx_Octet_Count_Good counter reaches half of the maximum value or the maximum value.	R/W
b1	RXGBOCTIM	MMC Receive Good Bad Octet Counter Interrupt Mask Setting this bit masks the interrupt generated when the Rx_Octet_Count_Good_Bad counter reaches half of the maximum value or the maximum value.	R/W
b0	RXGBFRMIM	MMC Receive Good Bad Frame Counter Interrupt Mask Setting this bit masks the interrupt generated when the Rx_Frames_Count_Good_Bad counter reaches half of the maximum value or the maximum value.	R/W

6.4.24 MMC_Transmit_Interrupt_Mask — MMC Transmit Interrupt Mask Register

Address: GMAC1: 4400 0110h
GMAC2: 4400 2110h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	TXOSIZEGFIM	TXVLANGFIM	TXPAUSFIM	TXEXDEFFIM	TXGFRMIM	TXGOCTIM	TXCARERFIM	TXEXCOLFIM	TXLATCOLFIM	TXDEFFIM
Value after reset	X	X	X	X	X	X	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TXMCOLGFIM	TXSCOLGFIM	TXUFLOWERFIM	TXBCGBFIM	TXMCGBFIM	TXUCGBFIM	TX1024TMAXOCTGBFIM	TX512T1023OCTGBFIM	TX256T511OCTGBFIM	TX128T255OCTGBFIM	TX65T127OCTGBFIM	TX64OCTGBFIM	TXMCGFIM	TXBCGFIM	TXGBFRMIM	TXGBOCTIM
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6.27 MMC_Transmit_Interrupt_Mask Register Contents (1/2)

Bit Position	Bit Name	Function	R/W
b31 to b26	Reserved	Reserved	R/W
b25	TXOSIZEGFIM	MMC Transmit Oversize Good Frame Counter Interrupt Mask Setting this bit masks the interrupt generated when the Tx_OSize_Frames_Good counter reaches half of the maximum value or the maximum value.	R/W
b24	TXVLANGFIM	MMC Transmit VLAN Good Frame Counter Interrupt Mask Setting this bit masks the interrupt generated when the Tx_VLAN_Frames_Good counter reaches half of the maximum value or the maximum value.	R/W
b23	TXPAUSFIM	MMC Transmit Pause Frame Counter Interrupt Mask Setting this bit masks the interrupt generated when the Tx_Pause_Frames counter reaches half of the maximum value or the maximum value.	R/W
b22	TXEXDEFFIM	MMC Transmit Excessive Deferral Frame Counter Interrupt Mask Setting this bit masks the interrupt generated when the Tx_Excessive_Deferral_Error counter reaches half of the maximum value or the maximum value.	R/W
b21	TXGFRMIM	MMC Transmit Good Frame Counter Interrupt Mask Setting this bit masks the interrupt generated when the Tx_Frame_Count_Good counter reaches half of the maximum value or the maximum value.	R/W
b20	TXGOCTIM	MMC Transmit Good Octet Counter Interrupt Mask Setting this bit masks the interrupt generated when the Tx_Octet_Count_Good counter reaches half of the maximum value or the maximum value.	R/W
b19	TXCARERFIM	MMC Transmit Carrier Error Frame Counter Interrupt Mask Setting this bit masks the interrupt generated when the Tx_Carrier_Error_Frames counter reaches half of the maximum value or the maximum value.	R/W
b18	TXEXCOLFIM	MMC Transmit Excessive Collision Frame Counter Interrupt Mask Setting this bit masks the interrupt generated when the Tx_Excessive_Collision_Frames counter reaches half of the maximum value or the maximum value.	R/W
b17	TXLATCOLFIM	MMC Transmit Late Collision Frame Counter Interrupt Mask Setting this bit masks the interrupt generated when the Tx_Late_Collision_Frames counter reaches half of the maximum value or the maximum value.	R/W
b16	TXDEFFIM	MMC Transmit Deferred Frame Counter Interrupt Mask Setting this bit masks the interrupt generated when the Tx_Deferred_Frames counter reaches half of the maximum value or the maximum value.	R/W
b15	TXMCOLGFIM	MMC Transmit Multiple Collision Good Frame Counter Interrupt Mask Setting this bit masks the interrupt generated when the Tx_Multiple_Collision_Good_Frames counter reaches half of the maximum value or the maximum value.	R/W

Table 6.27 MMC_Transmit_Interrupt_Mask Register Contents (2/2)

Bit Position	Bit Name	Function	R/W
b14	TXSCOLGFIM	MMC Transmit Single Collision Good Frame Counter Interrupt Mask Setting this bit masks the interrupt generated when the Tx_Single_Collision_Good_Frames counter reaches half of the maximum value or the maximum value.	R/W
b13	TXUFLOWERFIM	MMC Transmit Underflow Error Frame Counter Interrupt Mask Setting this bit masks the interrupt generated when the Tx_Underflow_Error_Frames counter reaches half of the maximum value or the maximum value.	R/W
b12	TXBCGBFIM	MMC Transmit Broadcast Good Bad Frame Counter Interrupt Mask Setting this bit masks the interrupt generated when the Tx_Broadcast_Frames_Good_Bad counter reaches half of the maximum value or the maximum value.	R/W
b11	TXMCGBFIM	MMC Transmit Multicast Good Bad Frame Counter Interrupt Mask Setting this bit masks the interrupt generated when the Tx_Multicast_Frames_Good_Bad counter reaches half of the maximum value or the maximum value.	R/W
b10	TXUCGBFIM	MMC Transmit Unicast Good Bad Frame Counter Interrupt Mask Setting this bit masks the interrupt generated when the Tx_Unicast_Frames_Good_Bad counter reaches half of the maximum value or the maximum value.	R/W
b9	TX1024TMAXOCTGBFIM	MMC Transmit 1024 to Maximum Octet Good Bad Frame Counter Interrupt Mask Setting this bit masks the interrupt generated when the Tx_1024ToMaxOctets_Frames_Good_Bad counter reaches half of the maximum value or the maximum value.	R/W
b8	TX512T1023OCTGBFIM	MMC Transmit 512 to 1023 Octet Good Bad Frame Counter Interrupt Mask Setting this bit masks the interrupt generated when the Tx_512To1023Octets_Frames_Good_Bad counter reaches half of the maximum value or the maximum value.	R/W
b7	TX256T511OCTGBFIM	MMC Transmit 256 to 511 Octet Good Bad Frame Counter Interrupt Mask Setting this bit masks the interrupt generated when the Tx_256To511Octets_Frames_Good_Bad counter reaches half of the maximum value or the maximum value.	R/W
b6	TX128T255OCTGBFIM	MMC Transmit 128 to 255 Octet Good Bad Frame Counter Interrupt Mask Setting this bit masks the interrupt generated when the Tx_128To255Octets_Frames_Good_Bad counter reaches half of the maximum value or the maximum value.	R/W
b5	TX65T127OCTGBFIM	MMC Transmit 65 to 127 Octet Good Bad Frame Counter Interrupt Mask Setting this bit masks the interrupt generated when the Tx_65To127Octets_Frames_Good_Bad counter reaches half of the maximum value or the maximum value.	R/W
b4	TX64OCTGBFIM	MMC Transmit 64 Octet Good Bad Frame Counter Interrupt Mask Setting this bit masks the interrupt generated when the Tx_64Octets_Frames_Good_Bad counter reaches half of the maximum value or the maximum value.	R/W
b3	TXMCGFIM	MMC Transmit Multicast Good Frame Counter Interrupt Mask Setting this bit masks the interrupt generated when the Tx_Multicast_Frames_Good counter reaches half of the maximum value or the maximum value.	R/W
b2	TXBCGFIM	MMC Transmit Broadcast Good Frame Counter Interrupt Mask Setting this bit masks the interrupt generated when the Tx_Broadcast_Frames_Good counter reaches half of the maximum value or the maximum value.	R/W
b1	TXGBFRMIM	MMC Transmit Good Bad Frame Counter Interrupt Mask Setting this bit masks the interrupt generated when the Tx_Frame_Count_Good_Bad counter reaches half of the maximum value or the maximum value.	R/W
b0	TXGBOCTIM	MMC Transmit Good Bad Octet Counter Interrupt Mask Setting this bit masks the interrupt generated when the Tx_Octet_Count_Good_Bad counter reaches half of the maximum value or the maximum value.	R/W

6.4.25 Tx_Octet_Count_Good_Bad — Transmit Octet Count for Good and Bad Frames

Address: GMAC1: 4400 0114h
GMAC2: 4400 2114h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TXOCTGB															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TXOCTGB															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6.28 Tx_Octet_Count_Good_Bad Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	TXOCTGB	This field indicates the number of bytes transmitted in good and bad frames exclusive of preamble and retried bytes.	R

6.4.26 Tx_Frame_Count_Good_Bad — Transmit Frame Count for Good and Bad Frames

Address: GMAC1: 4400 0118h
GMAC2: 4400 2118h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TXFRMGB															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TXFRMGB															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6.29 Tx_Frame_Count_Good_Bad Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	TXFRMGB	This field indicates the number of good and bad frames transmitted, exclusive of retried frames	R

6.4.27 Tx_Broadcast_Frames_Good — Transmit Frame Count for Good Broadcast Frames

Address: GMAC1: 4400 011Ch
GMAC2: 4400 211Ch

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TXBCASTG															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TXBCASTG															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6.30 Tx_Broadcast_Frames_Good Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	TXBCASTG	This field indicates the number of transmitted good broadcast frames.	R

6.4.28 Tx_Multicast_Frames_Good — Transmit Frame Count for Good Multicast Frames

Address: GMAC1: 4400 0120h
GMAC2: 4400 2120h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TXMCASTG															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TXMCASTG															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6.31 Tx_Multicast_Frames_Good Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	TXMCASTG	This field indicates the number of transmitted good multicast frames.	R

6.4.29 Tx_64Octets_Frames_Good_Bad — Transmit Octet Count for Good and Bad 64 Byte Frames

Address: GMAC1: 4400 0124h
GMAC2: 4400 2124h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TX64OCTGB															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TX64OCTGB															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6.32 Tx_64Octets_Frames_Good_Bad Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	TX64OCTGB	This field indicates the number of transmitted good and bad frames with length of 64 bytes, exclusive of preamble and retried frames.	R

6.4.30 Tx_65To127Octets_Frames_Good_Bad — Transmit Octet Count for Good and Bad 65 to 127 Bytes Frames

Address: GMAC1: 4400 0128h
GMAC2: 4400 2128h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TX65_127OCTGB															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TX65_127OCTGB															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6.33 Tx_65To127Octets_Frames_Good_Bad Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	TX65_127OCTGB	This field indicates the number of transmitted good and bad frames with length between 65 and 127 (inclusive) bytes, exclusive of preamble and retried frames.	R

6.4.31 Tx_128To255Octets_Frames_Good_Bad — Transmit Octet Count for Good and Bad 128 to 255 Bytes Frames

Address: GMAC1: 4400 012Ch
GMAC2: 4400 212Ch

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TX128_255OCTGB															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TX128_255OCTGB															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6.34 Tx_128To255Octets_Frames_Good_Bad Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	TX128_255OCTGB	This field indicates the number of transmitted good and bad frames with length between 128 and 255 (inclusive) bytes, exclusive of preamble and retried frames.	R

6.4.32 Tx_256To511Octets_Frames_Good_Bad — Transmit Octet Count for Good and Bad 256 to 511 Bytes Frames

Address: GMAC1: 4400 0130h
GMAC2: 4400 2130h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TX256_511OCTGB															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TX256_511OCTGB															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6.35 Tx_256To511Octets_Frames_Good_Bad Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	TX256_511OCTGB	This field indicates the number of transmitted good and bad frames with length between 256 and 511 (inclusive) bytes, exclusive of preamble and retried frames.	R

6.4.33 Tx_512To1023Octets_Frames_Good_Bad — Transmit Octet Count for Good and Bad 512 to 1023 Bytes Frames

Address: GMAC1: 4400 0134h
GMAC2: 4400 2134h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TX512_1023OCTGB															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TX512_1023OCTGB															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6.36 Tx_512To1023Octets_Frames_Good_Bad Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	TX512_1023OCTGB	This field indicates the number of transmitted good and bad frames with length between 512 and 1,023 (inclusive) bytes, exclusive of preamble and retried frames.	R

6.4.34 Tx_1024ToMaxOctets_Frames_Good_Bad — Transmit Octet Count for Good and Bad 1024 to Maxsize Bytes Frames

Address: GMAC1: 4400 0138h
GMAC2: 4400 2138h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TX1024_MAXOCTGB															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TX1024_MAXOCTGB															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6.37 Tx_1024ToMaxOctets_Frames_Good_Bad Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	TX1024_MAXOCTGB	This field indicates the number of good and bad frames transmitted with length between 1,024 and maxsize (inclusive) bytes, exclusive of preamble and retried frames.	R

6.4.35 Tx_Unicast_Frames_Good_Bad — Transmit Frame Count for Good and Bad Unicast Frames

Address: GMAC1: 4400 013Ch
GMAC2: 4400 213Ch

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TXUCASTGB															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TXUCASTGB															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6.38 Tx_Unicast_Frames_Good_Bad Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	TXUCASTGB	This field indicates the number of transmitted good and bad unicast frames.	R

6.4.36 Tx_Multicast_Frames_Good_Bad — Transmit Frame Count for Good and Bad Multicast Frames

Address: GMAC1: 4400 0140h
GMAC2: 4400 2140h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TXMCASTGB															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TXMCASTGB															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6.39 Tx_Multicast_Frames_Good_Bad Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	TXMCASTGB	This field indicates the number of transmitted good and bad multicast frames.	R

6.4.37 Tx_Broadcast_Frames_Good_Bad — Transmit Frame Count for Good and Bad Broadcast Frames

Address: GMAC1: 4400 0144h
GMAC2: 4400 2144h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TXBCASTGB															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TXBCASTGB															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6.40 Tx_Broadcast_Frames_Good_Bad Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	TXBCASTGB	This field indicates the number of transmitted good and bad broadcast frames.	R

6.4.38 Tx_Underflow_Error_Frames — Transmit Frame Count for Underflow Error Frames

Address: GMAC1: 4400 0148h
GMAC2: 4400 2148h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TXUNDRFLW															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6.41 Tx_Underflow_Error_Frames Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b16	Reserved	Reserved	R
b15 to b0	TXUNDRFLW	This field indicates the number of frames aborted because of frame underflow error.	R

6.4.39 Tx_Single_Collision_Good_Frames — Transmit Frame Count for Frames Transmitted after Single Collision

Address: GMAC1: 4400 014Ch
GMAC2: 4400 214Ch

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TXSNGLCOLG															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6.42 Tx_Single_Collision_Good_Frames Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b16	Reserved	Reserved	R
b15 to b0	TXSNGLCOLG	This field indicates the number of successfully transmitted frames after a single collision in the half-duplex mode.	R

6.4.40 Tx_Multiple_Collision_Good_Frames — Transmit Frame Count for Frames Transmitted after Multiple Collision

Address: GMAC1: 4400 0150h
GMAC2: 4400 2150h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TXMULTCOLG															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6.43 Tx_Multiple_Collision_Good_Frames Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b16	Reserved	Reserved	R
b15 to b0	TXMULTCOLG	This field indicates the number of successfully transmitted frames after multiple collisions in the half-duplex mode.	R

6.4.41 Tx_Deferred_Frames — Transmit Frame Count for Deferred Frames

Address: GMAC1: 4400 0154h
GMAC2: 4400 2154h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TXDEFRD															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6.44 Tx_Deferred_Frames Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b16	Reserved	Reserved	R
b15 to b0	TXDEFRD	This field indicates the number of successfully transmitted frames after a deferral in the half-duplex mode.	R

6.4.42 Tx_Late_Collision_Frames — Transmit Frame Count for Late Collision Error Frames

Address: GMAC1: 4400 0158h
GMAC2: 4400 2158h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TXLATECOL															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6.45 Tx_Late_Collision_Frames Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b16	Reserved	Reserved	R
b15 to b0	TXLATECOL	This field indicates the number of frames aborted because of late collision error.	R

6.4.43 Tx_Excessive_Collision_Frames — Transmit Frame Count for Excessive Collision Error Frames

Address: GMAC1: 4400 015Ch
GMAC2: 4400 215Ch

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TXEXSCOL															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6.46 Tx_Excessive_Collision_Frames Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b16	Reserved	Reserved	R
b15 to b0	TXEXSCOL	This field indicates the number of frames aborted because of excessive (16) collision error.	R

6.4.44 Tx_Carrier_Error_Frames — Transmit Frame Count for Carrier Sense Error Frames

Address: GMAC1: 4400 0160h
GMAC2: 4400 2160h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TXCARR															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6.47 Tx_Carrier_Error_Frames Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b16	Reserved	Reserved	R
b15 to b0	TXCARR	This field indicates the number of frames aborted because of carrier sense error (no carrier or loss of carrier).	R

6.4.45 Tx_Octet_Count_Good — Transmit Octet Count for Good Frames

Address: GMAC1: 4400 0164h
GMAC2: 4400 2164h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TXOCTG															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TXOCTG															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6.48 Tx_Octet_Count_Good Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	TXOCTG	This field indicates the number of bytes transmitted, exclusive of preamble, in good frames.	R

6.4.46 Tx_Frame_Count_Good — Transmit Frame Count for Good Frames

Address: GMAC1: 4400 0168h
GMAC2: 4400 2168h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TXFRMG															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TXFRMG															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6.49 Tx_Frame_Count_Good Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	TXFRMG	This field indicates the number of transmitted good frames, exclusive of preamble.	R

6.4.47 Tx_Excessive_Deferral_Error — Transmit Frame Count for Excessive Deferral Error Frames

Address: GMAC1: 4400 016Ch
GMAC2: 4400 216Ch

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TXEXSDEF															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6.50 Tx_Excessive_Deferral_Error Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b16	Reserved	Reserved	R
b15 to b0	TXEXSDEF	This field indicates the number of frames aborted because of excessive deferral error, that is, frames deferred for more than two max sized frame times.	R

6.4.48 Tx_Pause_Frames — Transmit Frame Count for Good PAUSE Frames

Address: GMAC1: 4400 0170h
GMAC2: 4400 2170h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TXPAUSE															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6.51 Tx_Pause_Frames Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b16	Reserved	Reserved	R
b15 to b0	TXPAUSE	This field indicates the number of transmitted good PAUSE frames.	R

6.4.49 Tx_VLAN_Frames_Good — Transmit Frame Count for Good VLAN Frames

Address: GMAC1: 4400 0174h
GMAC2: 4400 2174h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TXVLANG															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TXVLANG															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6.52 Tx_VLAN_Frames_Good Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	TXVLANG	This register maintains the number of transmitted good VLAN frames, exclusive of retried frames.	R

6.4.50 Tx_OSize_Frames_Good — Transmit Frame Count for Good Oversize Frames

Address: GMAC1: 4400 0178h
GMAC2: 4400 2178h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TXOSIZG															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6.53 Tx_OSize_Frames_Good Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b16	Reserved	Reserved	R
b15 to b0	TXOSIZG	This field indicates the number of frames transmitted without errors and with length greater than the maxsize (1,518 or 1,522 bytes for VLAN tagged frames; 2000 bytes if enabled in bit 27 of MAC Configuration Register (MAC_Configuration)).	R

6.4.51 Rx_Frames_Count_Good_Bad — Receive Frame Count for Good and Bad Frames

Address: GMAC1: 4400 0180h
GMAC2: 4400 2180h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	RXFRMGB															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RXFRMGB															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6.54 Rx_Frames_Count_Good_Bad Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	RXFRMGB	This field indicates the number of received good and bad frames.	R

6.4.52 Rx_Octet_Count_Good_Bad — Receive Octet Count for Good and Bad Frames

Address: GMAC1: 4400 0184h
GMAC2: 4400 2184h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	RXOCTGB															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RXOCTGB															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6.55 Rx_Octet_Count_Good_Bad Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	RXOCTGB	This field indicates the number of bytes received, exclusive of preamble, in good and bad frames.	R

6.4.53 Rx_Octet_Count_Good — Receive Octet Count for Good Frames

Address: GMAC1: 4400 0188h
GMAC2: 4400 2188h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	RXOCTG															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RXOCTG															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6.56 Rx_Octet_Count_Good Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	RXOCTG	This field indicates the number of bytes received, exclusive of preamble, only in good frames.	R

6.4.54 Rx_Broadcast_Frames_Good — Receive Frame Count for Good Broadcast Frames

Address: GMAC1: 4400 018Ch
GMAC2: 4400 218Ch

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	RXBCASTG															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RXBCASTG															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6.57 Rx_Broadcast_Frames_Good Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	RXBCASTG	This field indicates the number of received good broadcast frames.	R

6.4.55 Rx_Multicast_Frames_Good — Receive Frame Count for Good Multicast Frames

Address: GMAC1: 4400 0190h
GMAC2: 4400 2190h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	RXMCASTG															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RXMCASTG															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6.58 Rx_Multicast_Frames_Good Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	RXMCASTG	This field indicates the number of received good multicast frames.	R

6.4.56 Rx_CRC_Error_Frames — Receive Frame Count for CRC Error Frames

Address: GMAC1: 4400 0194h
GMAC2: 4400 2194h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RXCRCERR															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6.59 Rx_CRC_Error_Frames Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b16	Reserved	Reserved	R
b15 to b0	RXCRCERR	This field indicates the number of frames received with CRC error.	R

6.4.57 Rx_Alignment_Error_Frames — Receive Frame Count for Alignment Error Frames

Address: GMAC1: 4400 0198h
GMAC2: 4400 2198h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RXALGNERR															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6.60 Rx_Alignment_Error_Frames Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b16	Reserved	Reserved	R
b15 to b0	RXALGNERR	This field indicates the number of frames received with alignment (dribble) error. This field is valid only in the 10 or 100 Mbps mode.	R

6.4.58 Rx_Runt_Error_Frames — Receive Frame Count for Runt Error Frames

Address: GMAC1: 4400 019Ch
GMAC2: 4400 219Ch

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RXRUNTERR															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6.61 Rx_Runt_Error_Frames Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b16	Reserved	Reserved	R
b15 to b0	RXRUNTERR	This field indicates the number of frames received with runt error (< 64 bytes and CRC error).	R

6.4.59 Rx_Jabber_Error_Frames — Receive Frame Count for Jabber Error Frames

Address: GMAC1: 4400 01A0h
GMAC2: 4400 21A0h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RXJABERR															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6.62 Rx_Jabber_Error_Frames Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b16	Reserved	Reserved	R
b15 to b0	RXJABERR	This field indicates the number of giant frames received with length (including CRC) greater than 1,518 bytes (1,522 bytes for VLAN tagged) and with CRC error. If Jumbo Frame mode is enabled, then frames of length greater than 9,018 bytes (9,022 for VLAN tagged) are considered as giant frames.	R

6.4.60 Rx_Undersize_Frames_Good — Receive Frame Count for Undersize Frames

Address: GMAC1: 4400 01A4h
GMAC2: 4400 21A4h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RXUNDERSZG															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6.63 Rx_Undersize_Frames_Good Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b16	Reserved	Reserved	R
b15 to b0	RXUNDERSZG	This field indicates the number of frames received with length less than 64 bytes and without errors.	R

6.4.61 Rx_Oversize_Frames_Good — Receive Frame Count for Oversize Frames

Address: GMAC1: 4400 01A8h
GMAC2: 4400 21A8h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RXOVERSZG															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6.64 Rx_Oversize_Frames_Good Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b16	Reserved	Reserved	R
b15 to b0	RXOVERSZG	This field indicates the number of frames received without errors, with length greater than the maxsize (1,518 or 1,522 for VLAN tagged frames; 2,000 bytes if enabled in bit 27 of MAC Configuration Register (MAC_Configuration)).	R

6.4.62 Rx_64Octets_Frames_Good_Bad — Receive Frame Count for Good and Bad 64 Byte Frames

Address: GMAC1: 4400 01ACh
GMAC2: 4400 21ACh

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	RX64OCTGB															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RX64OCTGB															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6.65 Rx_64Octets_Frames_Good_Bad Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	RX64OCTGB	This field indicates the number of received good and bad frames with length 64 bytes, exclusive of preamble.	R

6.4.63 Rx_65To127Octets_Frames_Good_Bad — Receive Frame Count for Good and Bad 65 to 127 Bytes Frames

Address: GMAC1: 4400 01B0h
GMAC2: 4400 21B0h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	RX65_127OCTGB															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RX65_127OCTGB															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6.66 Rx_65To127Octets_Frames_Good_Bad Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	RX65_127OCTGB	This field indicates the number of received good and bad frames received with length between 65 and 127 (inclusive) bytes, exclusive of preamble.	R

6.4.64 Rx_128To255Octets_Frames_Good_Bad — Receive Frame Count for Good and Bad 128 to 255 Bytes Frames

Address: GMAC1: 4400 01B4h
GMAC2: 4400 21B4h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	RX128_255OCTGB															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RX128_255OCTGB															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6.67 Rx_128To255Octets_Frames_Good_Bad Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	RX128_255OCTGB	This field indicates the number of received good and bad frames with length between 128 and 255 (inclusive) bytes, exclusive of preamble.	R

6.4.65 Rx_256To511Octets_Frames_Good_Bad — Receive Frame Count for Good and Bad 256 to 511 Bytes Frames

Address: GMAC1: 4400 01B8h
GMAC2: 4400 21B8h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	RX256_511OCTGB															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RX256_511OCTGB															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6.68 Rx_256To511Octets_Frames_Good_Bad Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	RX256_511OCTGB	This field indicates the number of received good and bad frames with length between 256 and 511 (inclusive) bytes, exclusive of preamble.	R

6.4.66 Rx_512To1023Octets_Frames_Good_Bad — Receive Frame Count for Good and Bad 512 to 1,023 Bytes Frames

Address: GMAC1: 4400 01BCh
GMAC2: 4400 21BCh

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	RX512_1023OCTGB															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RX512_1023OCTGB															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6.69 Rx_512To1023Octets_Frames_Good_Bad Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	RX512_1023OCTGB	This field indicates the number of received good and bad frames with length between 512 and 1,023 (inclusive) bytes, exclusive of preamble.	R

6.4.67 Rx_1024ToMaxOctets_Frames_Good_Bad — Receive Frame Count for Good and Bad 1,024 to Maxsize Bytes Frames

Address: GMAC1: 4400 01C0h
GMAC2: 4400 21C0h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	RX1024_MAXOCTGB															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RX1024_MAXOCTGB															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6.70 Rx_1024ToMaxOctets_Frames_Good_Bad Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	RX1024_MAXOCTGB	This field indicates the number of received good and bad frames with length between 1,024 and maxsize (inclusive) bytes, exclusive of preamble and retried frames.	R

6.4.68 Rx_Unicast_Frames_Good — Receive Frame Count for Good Unicast Frames

Address: GMAC1: 4400 01C4h
GMAC2: 4400 21C4h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	RXUCASTG															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RXUCASTG															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6.71 Rx_Unicast_Frames_Good Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	RXUCASTG	This field indicates the number of received good unicast frames.	R

6.4.69 Rx_Length_Error_Frames — Receive Frame Count for Length Error Frames

Address: GMAC1: 4400 01C8h
GMAC2: 4400 21C8h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RXLENERR															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6.72 Rx_Length_Error_Frames Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b16	Reserved	Reserved	R
b15 to b0	RXLENERR	This field indicates the number of frames received with length error (Length type field not equal to frame size) for all frames with valid length field.	R

6.4.70 Rx_Out_Of_Range_Type_Frames — Receive Frame Count for Out of Range Frames

Address: GMAC1: 4400 01CCh
GMAC2: 4400 21CCh

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RXOUTOFRNG															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6.73 Rx_Out_Of_Range_Type_Frames Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b16	Reserved	Reserved	R
b15 to b0	RXOUTOFRNG	This field indicates the number of received frames with length field not equal to the valid frame size (greater than 1,500 but less than 1,536).	R

6.4.71 Rx_Pause_Frames — Receive Frame Count for PAUSE Frames

Address: GMAC1: 4400 01D0h
GMAC2: 4400 21D0h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RXPAUSEFRM															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6.74 Rx_Pause_Frames Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b16	Reserved	Reserved	R
b15 to b0	RXPAUSEFRM	This field indicates the number of received good and valid PAUSE frames.	R

6.4.72 Rx_FIFO_Overflow_Frames — Receive Frame Count for FIFO Overflow Frames

Address: GMAC1: 4400 01D4h
GMAC2: 4400 21D4h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RXFIFOOVFL															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6.75 Rx_FIFO_Overflow_Frames Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b16	Reserved	Reserved	R
b15 to b0	RXFIFOOVFL	This field indicates the number of received frames missed because of FIFO overflow.	R

6.4.73 Rx_VLAN_Frames_Good_Bad — Receive Frame Count for Good and Bad VLAN Frames

Address: GMAC1: 4400 01D8h
GMAC2: 4400 21D8h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	RXVLANFRGB															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RXVLANFRGB															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6.76 Rx_VLAN_Frames_Good_Bad Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	RXVLANFRGB	This field indicates the number of received good and bad VLAN frames.	R

6.4.74 Rx_Watchdog_Error_Frames — Receive Frame Count for Watchdog Error Frames

Address: GMAC1: 4400 01DCh
GMAC2: 4400 21DCh

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RXWDGERR															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6.77 Rx_Watchdog_Error_Frames Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b16	Reserved	Reserved	R
b15 to b0	RXWDGERR	This field indicates the number of frames received with error because of the watchdog timeout error (frames with more than 2,048 bytes or value programmed in Watchdog Timeout Register (WDog_Timeout)).	R

6.4.75 Rx_Receive_Error_Frames — Receive Frame Count for Receive Error Frames

Address: GMAC1: 4400 01E0h
GMAC2: 4400 21E0h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RXRCVERR															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6.78 Rx_Receive_Error_Frames Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b16	Reserved	Reserved	R
b15 to b0	RXRCVERR	This field indicates the number of frames received with error because of the GMII/MII RXER error or Frame Extension error on GMII.	R

6.4.76 Rx_Control_Frames_Good — Receive Frame Count for Good Control Frames

Address: GMAC1: 4400 01E4h
GMAC2: 4400 21E4h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	RXCTRLG															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RXCTRLG															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6.79 Rx_Control_Frames_Good Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	RXCTRLG	This field indicates the number of good control frames received.	R

6.4.77 Hash_Table_Reg[n] — Hash Table Register [n] (n = 0..7)

Address: GMAC1: 4400 0500h +4h × n
GMAC2: 4400 2500h +4h × n

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	HT[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	HT[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6.80 Hash_Table_Reg[n] Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	HT[n]	This field contains the [n]th 32 Bits (31:0) of the Hash table.	R/W

6.4.78 VLAN_Hash_Table_Reg — VLAN Hash Table Register

Address: GMAC1: 4400 0588h
GMAC2: 4400 2588h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	VLHT															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6.81 VLAN_Hash_Table_Reg Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b16	Reserved	Reserved	R/W
b15 to b0	VLHT	VLAN Hash Table This field contains the 16-bit VLAN Hash Table.	R/W

6.4.79 Timestamp_Control — Timestamp Control Register

Address: GMAC1: 4400 0700h
GMAC2: 4400 2700h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	ATSEN0	ATSFC	—	—	—	—	—	TSENMACADDR	SNAPTYPSEL	
Value after reset	X	X	X	X	X	X	0	0	X	X	X	X	X	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TSMSTRENA	TSEVNTENA	TSIPV4ENA	TSIPV6ENA	TSIPENA	TSVER2ENA	TSCTRLSSR	TSENALL	—	—	TSADDREG	TSTRIG	TSUPDT	TSINIT	TSCFUPDT	TSENA
Value after reset	0	0	1	0	0	0	0	0	X	X	0	0	0	0	0	0

Table 6.82 Timestamp_Control Register Contents (1/2)

Bit Position	Bit Name	Function	R/W
b31 to b26	Reserved	Reserved	R/W
b25	ATSEN0	Auxiliary Snapshot 0 Enable This field controls capturing the Auxiliary Snapshot Trigger 0. When this bit is set, the Auxiliary snapshot of event on MAC_TRIG[m] (GMAC1:m=1, GMAC2:m=2) input is enabled. When this bit is reset, the events on this input are ignored.	R/W
b24	ATSFC	Auxiliary Snapshot FIFO Clear When set, it resets the pointers of the Auxiliary Snapshot FIFO. This bit is cleared when the pointers are reset and the FIFO is empty. When this bit is high, auxiliary snapshots get stored in the FIFO.	R/W
b23 to b19	Reserved	Reserved	R/W
b18	TSENMACADDR	Enable MAC address for PTP Frame Filtering When set, the DA MAC address (that matches any MAC Address register) is used to filter the PTP frames when PTP is directly sent over Ethernet.	R/W
b17, b16	SNAPTYPSEL	Select PTP packets for Taking Snapshots These bits along with Bits 15 and 14 decide the set of PTP packet types for which snapshot needs to be taken.	R/W
b15	TSMSTRENA	Enable Snapshot for Messages Relevant to Master When set, the snapshot is taken only for the messages relevant to the master node. Otherwise, the snapshot is taken for the messages relevant to the slave node.	R/W
b14	TSEVNTENA	Enable Timestamp Snapshot for Event Messages When set, the timestamp snapshot is taken only for event messages (SYNC, Delay_Req, Pdelay_Req, or Pdelay_Resp). When reset, the snapshot is taken for all messages except Announce, Management, and Signaling.	R/W
b13	TSIPV4ENA	Enable Processing of PTP Frames Sent over IPv4-UDP When set, the GMAC receiver processes the PTP packets encapsulated in UDP over IPv4 packets. When this bit is clear, the GMAC ignores the PTP transported over UDP-IPv4 packets. This bit is set by default.	R/W
b12	TSIPV6ENA	Enable Processing of PTP Frames Sent Over IPv6-UDP When set, the GMAC receiver processes PTP packets encapsulated in UDP over IPv6 packets. When this bit is clear, the GMAC ignores the PTP transported over UDP-IPv6 packets.	R/W
b11	TSIPENA	Enable Processing of PTP over Ethernet Frames When set, the GMAC receiver processes the PTP packets encapsulated directly in the Ethernet frames. When this bit is clear, the GMAC ignores the PTP over Ethernet packets.	R/W
b10	TSVER2ENA	Enable PTP packet Processing for Version 2 Format When set, the PTP packets are processed using the IEEE 1588 version 2 format. Otherwise, the PTP packets are processed using the version 1 format.	R/W

Table 6.82 Timestamp_Control Register Contents (2/2)

Bit Position	Bit Name	Function	R/W
b9	TSCTRLSSR	Timestamp Digital or Binary Rollover Control When set, the Timestamp Low register rolls over after 0x3B9A_C9FF value (that is, 1 nanosecond accuracy) and increments the timestamp (High) seconds. When 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 the value of this bit.	R/W
b8	TSENALL	Enable Timestamp for All Frames When set, the timestamp snapshot is enabled for all frames received by the GMAC.	R/W
b7, b6	Reserved	Reserved	R/W
b5	TSADDREG	Addend Reg Update When set, the content of the Timestamp 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. GMAC1 is available only. Reserved at GMAC2.	R/W
b4	TSTRIG	Timestamp Interrupt Trigger Enable When set, the timestamp interrupt is generated when the System Time becomes greater than the value written in the Target Time register. This bit is reset after the generation of the Timestamp Trigger Interrupt. GMAC1 is available only. Reserved at GMAC2.	R/W
b3	TSUPDT	Timestamp Update When set, the system time is updated (added or subtracted) with the value specified in System Time – Seconds Update Register (System_Time_Seconds_Update) and System Time – Nanoseconds Update Register (System_time_Nanoseconds_Update). This bit should be read zero before updating it. This bit is reset when the update is completed in hardware. GMAC1 is available only. Reserved at GMAC2.	R/W
b2	TSINIT	Timestamp Initialize When set, the system time is initialized (overwritten) with the value specified in the System Time – Seconds Update Register (System_Time_Seconds_Update) and System Time – Nanoseconds Update Register (System_time_Nanoseconds_Update). This bit should be read zero before updating it. This bit is reset when the initialization is complete. GMAC1 is available only. Reserved at GMAC2.	R/W
b1	TSCFUPDT	Timestamp Fine or Coarse Update When set, this bit indicates that the system times update should be done using the fine update method. When reset, it indicates the system timestamp update should be done using the Coarse method. GMAC1 is available only. Reserved at GMAC2.	R/W
b0	TSENA	Timestamp Enable When set, the timestamp is added for the transmit and receive frames. When disabled, timestamp is not added for the transmit and receive frames and the Timestamp Generator is also suspended. You need to initialize the Timestamp (system time) after enabling this mode. On the receive side, the GMAC processes the IEEE 1588 frames only if this bit is set.	R/W

6.4.80 Sub_Second_Increment — Sub-Second Increment Register

Address: GMAC1: 4400 0704h
GMAC2: Reserved

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	SSINC							
Value after reset	X	X	X	X	X	X	X	X	0	0	0	0	0	0	0	0

Table 6.83 Sub_Second_Increment Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b8	Reserved	Reserved	R/W
b7 to b0	SSINC	Sub-second Increment Value The value programmed in this field is accumulated every clock cycle (of GMAC_PTP_REFCLK_I) with the contents of the sub-second register. For example, when PTP clock is 50 MHz (period is 20 ns), you should program 20 (0x14) when the System Time-Nanoseconds register has an accuracy of 1 ns (TSCTRLSSR bit is set). When TSCTRLSSR is clear, the Nanoseconds register has a resolution of ~0.465 ns. In this case, you should program a value of 43 (0x2B) that is derived by 20 ns/0.465.	R/W

6.4.81 System_Time_Seconds — System Time – Seconds Register

Address: GMAC1: 4400 0708h
GMAC2: Reserved

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TSS															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TSS															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6.84 System_Time_Seconds Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	TSS	Timestamp Second The value in this field indicates the current value in seconds of the System Time maintained by the GMAC.	R

6.4.82 System_Time_Nanoseconds — System Time – Nanoseconds Register

Address: GMAC1: 4400 070Ch
GMAC2: Reserved

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	TSSS														
Value after reset	X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TSSS															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6.85 System_Time_Nanoseconds Register Contents

Bit Position	Bit Name	Function	R/W
b31	Reserved	Reserved	R
b30 to b0	TSSS	Timestamp Sub Seconds The value in this field has the sub second representation of time, with an accuracy of 0.46 ns. When bit 9 (TSCTRLSSR) is set in Timestamp Control Register (Timestamp_Control), each bit represents 1 ns and the maximum value is 0x3B9A_C9FF, after which it rolls-over to zero.	R

6.4.83 System_Time_Seconds_Update — System Time – Seconds Update Register

Address: GMAC1: 4400 0710h
GMAC2: Reserved

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TSS															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TSS															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6.86 System_Time_Seconds_Update Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	TSS	Timestamp Second The value in this field indicates the time in seconds to be initialized or added to the system time.	R/W

6.4.84 System_Time_Nanoseconds_Update — System Time – Nanoseconds Update Register

Address: GMAC1: 4400 0714h
GMAC2: Reserved

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	ADDSUB	TSSS														
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TSSS															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6.87 System_Time_Nanoseconds_Update Register Contents

Bit Position	Bit Name	Function	R/W
b31	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.	R/W
b30 to b0	TSSS	Timestamp Sub Second The value in this field has the sub second representation of time, with an accuracy of 0.46 ns. When bit 9 (TSCTRLSSR) is set in Timestamp Control Register (Timestamp_Control), each bit represents 1 ns and the programmed value should not exceed 0x3B9A_C9FF.	R/W

6.4.85 Timestamp_Addend — Timestamp Addend Register

Address: GMAC1: 4400 0718h
GMAC2: Reserved

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TSAR															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TSAR															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6.88 Timestamp_Addend Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	TSAR	Timestamp Addend Register This field indicates the 32-bit time value to be added to the Accumulator to achieve time synchronization.	R/W

6.4.86 Target_Time_Seconds — Target Time Seconds Register

Address: GMAC1: 4400 071Ch
GMAC2: Reserved

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TSTR															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TSTR															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6.89 Target_Time_Seconds Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	TSTR	Target Time Seconds Register This register stores the time in seconds. When the timestamp value matches or exceeds both Target Timestamp registers, then based on Bits[6:5] of PPS Control Register (PPS_Control), the GMAC starts or stops the PPS signal output and generates an interrupt (if enabled).	R/W

6.4.87 Target_Time_Nanoseconds — Target Time Nanoseconds Register

Address: GMAC1: 4400 0720h
GMAC2: Reserved

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TRGTBUSY	TTSLO														
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TTSLO															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6.90 Target_Time_Nanoseconds Register Contents

Bit Position	Bit Name	Function	R/W
b31	TRGTBUSY	Target Time Register Busy The GMAC sets this bit when the PPSCMD field (Bits[3:0]) in PPS Control Register (PPS_Control) is programmed to 3'b010 or 3'b011. Programming the PPSCMD field to 3'b010 or 3'b011, instructs the GMAC to synchronize the Target Time Registers to the PTP clock domain. The GMAC clears this bit after synchronizing the Target Time Registers to the PTP clock domain. The application must not update the Target Time Registers when this bit is read as 1. Otherwise, the synchronization of the previous programmed time gets corrupted.	R
b30 to b0	TTSLO	Target Timestamp Low Register This register stores the time in (signed) nanoseconds. When the value of the timestamp matches the both Target Timestamp registers, then based on the TRGTMODSEL0 field (Bits [6:5]) in PPS Control Register (PPS_Control), the GMAC starts or stops the PPS signal output and generates an interrupt (if enabled). This value should not exceed 0x3B9A_C9FF when TSCTRLSSR is set in the Timestamp control register. The actual start or stop time of the PPS signal output may have an error margin up to one unit of sub-second increment value.	R/W

6.4.88 Timestamp_Status — Timestamp Status Register

Address: GMAC1: 4400 0728h
GMAC2: 4400 2728h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	ATSNS					ATSSTM	—	—	—	—	ATSSTN			
Value after reset	X	X	0	0	0	0	0	0	X	X	X	X	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	TSTRGTERR1	TSTARGT1	TSTRGTERR	AUXTSTRIG	TSTARGT	TSSOVF
Value after reset	X	X	X	X	X	X	X	X	X	X	0	0	0	0	0	0

Table 6.91 Timestamp_Status Register Contents (1/2)

Bit Position	Bit Name	Function	R/W
b31, b30	Reserved	Reserved	R
b29 to b25	ATSNS	Number of Auxiliary Timestamp Snapshots This field indicates the number of Snapshots available in the FIFO. A value equal to the selected depth of FIFO (4) indicates that the Auxiliary Snapshot FIFO is full. These bits are cleared (to 5'b00000) when the Auxiliary snapshot FIFO clear bit is set.	R
b24	ATSSTM	Auxiliary Timestamp Snapshot Trigger Missed This bit is set when the Auxiliary timestamp snapshot FIFO is full and external trigger was set. This indicates that the latest snapshot is not stored in the FIFO.	R
b23 to b20	Reserved	Reserved	R
b19 to b16	ATSSTN	Auxiliary Timestamp Snapshot Trigger Identifier These bits identify the Auxiliary trigger inputs for which the timestamp available in the Auxiliary Snapshot Register is applicable. When more than one bit is set at the same time, it means that corresponding auxiliary triggers were sampled at the same clock. These bits are applicable only if the number of Auxiliary snapshots is more than one. One bit is assigned for each trigger as shown in the following list: Bit 16: Auxiliary trigger 0 Bit 17: Auxiliary trigger 1 (Not used) Bit 18: Auxiliary trigger 2 (Not used) Bit 19: Auxiliary trigger 3 (Not used) The software can read this register to find the triggers that are set when the timestamp is taken.	R
b15 to b6	Reserved	Reserved	R
b5	TSTRGTERR1	Timestamp Target Time Error This bit is set when the target time, being programmed in PPS1 Target Time High Register (PPS1_Target_Time_Seconds) and PPS1 Target Time Low Register (PPS1_Target_Time_Nanoseconds), is already elapsed. This bit is cleared when read by the application. GMAC1 is available only. Reserved at GMAC2.	R
b4	TSTARGT1	Timestamp Target Time Reached for Target Time PPS1 When set, this bit indicates that the value of system time is greater than or equal to the value specified in PPS1 Target Time High Register (PPS1_Target_Time_Seconds) and PPS1 Target Time Low Register (PPS1_Target_Time_Nanoseconds). GMAC1 is available only. Reserved at GMAC2.	R
b3	TSTRGTERR	Timestamp Target Time Error This bit is set when the target time, being programmed in Target Time Registers, is already elapsed. This bit is cleared when read by the application. GMAC1 is available only. Reserved at GMAC2.	R

Table 6.91 Timestamp_Status Register Contents (2/2)

Bit Position	Bit Name	Function	R/W
b2	AUXTSTRIG	Auxiliary Timestamp Trigger Snapshot This bit is set high when the auxiliary snapshot is written to the FIFO.	R
b1	TSTARGT	Timestamp Target Time Reached When set, this bit indicates that the value of system time is greater or equal to the value specified in the Target Time Seconds Register (Target_Time_Seconds) and Target Time Nanoseconds Register (Target_Time_Nanoseconds). GMAC1 is available only. Reserved at GMAC2.	R
b0	TSSOVF	Timestamp Seconds Overflow When set, this bit indicates that the seconds value of the timestamp (when supporting version 2 format) has overflowed beyond 32'hFFFF_FFFF. GMAC1 is available only. Reserved at GMAC2.	R

6.4.89 PPS_Control — PPS Control Register

Address: GMAC1: 4400 072Ch
GMAC2: Reserved

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	TRGTMODSEL3		—	—	PPSCMD3			—	TRGTMODSEL2		—	—	PPSCMD2		
Value after reset	X	0	0	X	X	0	0	0	X	0	0	X	X	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	TRGTMODSEL1		—	—	PPSCMD1			—	TRGTMODSEL0		PPSEN0	PPSCTRL_PPSCMD			
Value after reset	X	0	0	X	X	0	0	0	X	0	0	0	0	0	0	0

Table 6.92 PPS_Control Register Contents (1/2)

Bit Position	Bit Name	Function	R/W
b31	Reserved	Reserved	R
b30, b29	TRGTMODSEL3	Target Time Register Mode for PPS3 Output This field is not used.	R
b28, b27	Reserved	Reserved	R
b26 to b24	PPSCMD3	Flexible PPS3 Output Control This field is not used.	R
b23	Reserved	Reserved	R
b22, b21	TRGTMODSEL2	Target Time Register Mode for PPS2 Output This field is not used.	R
b20, b19	Reserved	Reserved	R
b18 to b16	PPSCMD2	Flexible PPS2 Output Control This field is not used.	R
b15	Reserved	Reserved	R
b14, b13	TRGTMODSEL1	Target Time Register Mode for PPS1 Output This field indicates the Target Time registers mode for PPS1 output signal. This field is similar to the TRGTMODSEL0 field.	R/W
b12, b11	Reserved	Reserved	R
b10 to b8	PPSCMD1	Flexible PPS1 Output Control This field controls the flexible PPS1 output (MAC_PPS[1]) signal. This field is similar to PPSCMD0[2:0] in functionality.	R/W
b7	Reserved	Reserved	R
b6, b5	TRGTMODSEL0	Target Time Register Mode for PPS0 Output This field indicates the Target Time registers mode for PPS0 output signal: 2'b00: Indicates that the Target Time registers are programmed only for generating the interrupt event. 2'b01: Reserved 2'b10: Indicates that the Target Time registers are programmed for generating the interrupt event and starting or stopping the generation of the PPS0 output signal. 2'b11: Indicates that the Target Time registers are programmed only for starting or stopping the generation of the PPS0 output signal. No interrupt is asserted.	R/W
b4	PPSEN0	Flexible PPS Output Mode Enable When set low, Bits[3:0] function as PPSCTRL (backward compatible). When set high, Bits[3:0] function as PPSCMD.	R/W

Table 6.92 PPS_Control Register Contents (2/2)

Bit Position	Bit Name	Function	R/W
b3 to b0	PPSCTRL_PPSCMD	PPSCTRL0 or PPSCMD0 PPSCTRL0: PPS0 Output Frequency Control This field controls the frequency of the PPS0 output (MAC_PPS[0]) signal. The default value of PPSCTRL is 4'b0000, and the PPS output is 1 pulse (of width GMAC_PTP_REFCLK_I) every second. For other values of PPSCTRL, the PPS output becomes a generated clock of following frequencies: 4'b0001: The binary rollover is 2 Hz, and the digital rollover is 1 Hz. 4'b0010: The binary rollover is 4 Hz, and the digital rollover is 2 Hz. 4'b0011: The binary rollover is 8 Hz, and the digital rollover is 4 Hz. 4'b0100: The binary rollover is 16 Hz, and the digital rollover is 8 Hz. ... 4'b1111: The binary rollover is 32.768 kHz, and the digital rollover is 16.384 kHz. ***Note)*** In the binary rollover mode, the PPS output (MAC_PPS) 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 = 4'b0001, the PPS (1 Hz) has a low period of 537 ms and a high period of 463 ms • When PPSCTRL = 4'b0010, 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 = 4'b0011, 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 bits in the digital rollover mode in System Time – Nanoseconds Register (System_Time_Nanoseconds). PPSCMD0: Flexible PPS0 Output (MAC_PPS[0]) Control Programming these bits with a non-zero value instructs the GMAC to initiate an event. Once the command is transferred or synchronized to the PTP clock domain, these bits get cleared automatically. The Software should ensure that these bits are programmed only when they are all-zero. The following list describes the values of PPSCMD0: 4'b0000: No Command 4'b0001: START Single Pulse This command generates single pulse rising at the start point defined in Target Time Registers (Target_Time_Seconds and Target_Time_Nanoseconds) and of a duration defined in the PPS0 Width Register. 4'b0010: START Pulse Train This command generates the train of pulses rising at the start point defined in the Target Time Registers and of a duration defined in the PPS0 Width Register and repeated at interval defined in the PPS Interval Register. By default, the PPS pulse train is free-running unless stopped by "STOP Pulse train at time" or "STOP Pulse Train immediately" commands. 4'b0011: Cancel START This command cancels the START Single Pulse and START Pulse Train commands if the system time has not crossed the programmed start time. 4'b0100: STOP Pulse train at time This command stops the train of pulses initiated by the START Pulse Train command (PPSCMD = 4'b0010) after the time programmed in the Target Time registers elapses. 4'b0101: STOP Pulse Train immediately This command immediately stops the train of pulses initiated by the START Pulse Train command (PPSCMD = 4'b0010). 4'b0110: Cancel STOP Pulse train This command cancels the STOP pulse train at time command if the programmed stop time has not elapsed. The PPS pulse train becomes free-running on the successful execution of this command. 4'b0111-4'b1111: Reserved	R/W

6.4.90 Auxiliary_Timestamp_Nanoseconds — Auxiliary Timestamp – Nanoseconds Register

Address: GMAC1: 4400 0730h
GMAC2: 4400 2730h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	AUXTSLO														
Value after reset	X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	AUXTSLO															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6.93 Auxiliary_Timestamp_Nanoseconds Register Contents

Bit Position	Bit Name	Function	R/W
b31	Reserved	Reserved	R
b30 to b0	AUXTSLO	Contains the lower 32 bits (nano seconds field) of the auxiliary timestamp.	R

6.4.91 Auxiliary_Timestamp_Seconds — Auxiliary Timestamp – Seconds Register

Address: GMAC1: 4400 0734h
GMAC2: 4400 2734h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	AUXTSHI															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	AUXTSHI															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6.94 Auxiliary_Timestamp_Seconds Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	AUXTSHI	Contains the upper 32 bits (Seconds field) of the auxiliary timestamp.	R

6.4.92 PPS0_Interval — PPS0 Interval Register

Address: GMAC1: 4400 0760h
GMAC2: Reserved

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	PPSINT															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PPSINT															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6.95 PPS0_Interval Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	PPSINT	PPS0 Output Signal Interval These bits store the interval between the rising edges of PPS0 signal output in terms of units of sub-second increment value. You need to program one value less than the required interval. For example, if the PTP reference clock is 50 MHz (period of 20 ns), and desired interval between rising edges of PPS0 signal output is 100 ns (that is, five units of sub-second increment value), then you should program value 4 (5 − 1) in this register.	R/W

6.4.93 PPS0_Width — PPS0 Width Register

Address: GMAC1: 4400 0764h
GMAC2: Reserved

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	PPSWIDTH															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PPSWIDTH															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6.96 PPS0_Width Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	PPSWIDTH	PPS0 Output Signal Width These bits store the width between the rising edge and corresponding falling edge of the PPS0 signal output in terms of units of sub second increment value. You need to program one value less than the required interval. For example, if PTP reference clock is 50 MHz (period of 20 ns), and desired width between the rising and corresponding falling edges of PPS0 signal output is 80 ns (that is, four units of sub second increment value), then you should program value 3 (4 − 1) in this register. ***Note*)** The value programmed in this register must be lesser than the value programmed in PPS0 Interval Register (PPS0_Interval).	R/W

6.4.94 PPS1_Target_Time_Seconds — PPS1 Target Time Seconds Register

Address: GMAC1: 4400 0780h
GMAC2: Reserved

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TSTRH1															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TSTRH1															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6.97 PPS1_Target_Time_Seconds Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	TSTRH1	PPS1 Target Time Seconds Register This register stores the time in seconds. When the timestamp value matches or exceeds both Target Timestamp registers, then based on Bits [14:13], TRGTMODSEL1, of PPS Control Register (PPS_Control), the GMAC starts or stops the PPS signal output and generates an interrupt (if enabled).	R/W

6.4.95 PPS1_Target_Time_Nanoseconds — PPS1 Target Time Nanoseconds Register

Address: GMAC1: 4400 0784h
GMAC2: Reserved

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TRGTBUSY1	TTSL1														
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TTSL1															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6.98 PPS1_Target_Time_Nanoseconds Register Contents

Bit Position	Bit Name	Function	R/W
b31	TRGTBUSY1	PPS1 Target Time Register Busy The GMAC sets this bit when the PPSCMD1 field (Bits [10:8]) in PPS Control Register (PPS_Control) is programmed to 3'b010 or 3'b011. Programming the PPSCMD1 field to 3'b010 or 3'b011 instructs the GMAC to synchronize the Target Time Registers to the PTP clock domain. The GMAC clears this bit after synchronizing the Target Time Registers to the PTP clock domain. The application must not update the Target Time Registers when this bit is read as 1. Otherwise, the synchronization of the previous programmed time gets corrupted.	R/W
b30 to b0	TTSL1	Target Time Low for PPS1 Register This register stores the time in (signed) nanoseconds. When the value of the timestamp matches the both Target Timestamp registers, then based on the TRGTMODSEL1 field (Bits [14:13]) in PPS Control Register (PPS_Control), the GMAC starts or stops the PPS signal output and generates an interrupt (if enabled). This value should not exceed 0x3B9A_C9FF when Bit 9 (TSCTRLSSR) is set in Timestamp Control Register (Timestamp_Control). The actual start or stop time of the PPS signal output may have an error margin up to one unit of sub-second increment value.	R/W

6.4.96 PPS1_Interval — PPS1 Interval Register

Address: GMAC1: 4400 0788h
GMAC2: Reserved

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	PPSINT															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PPSINT															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6.99 PPS1_Interval Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	PPSINT	PPS1 Output Signal Interval These bits store the interval between the rising edges of PPS1 signal output in terms of units of sub second increment value. You need to program one value less than the required interval. For example, if the PTP reference clock is 50 MHz (period of 20 ns), and desired interval between rising edges of PPS1 signal output is 100 ns (that is, five units of sub second increment value), then you should program value 4 (5 − 1) in this register.	R/W

6.4.97 PPS1_Width — PPS1 Width Register

Address: GMAC1: 4400 078Ch
GMAC2: Reserved

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	PPSWIDTH															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PPSWIDTH															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6.100 PPS1_Width Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	PPSWIDTH	PPS1 Output Signal Width These bits store the width between the rising edge and corresponding falling edge of the PPS1 signal output in terms of units of sub second increment value. You need to program one value less than the required interval. For example, if PTP reference clock is 50 MHz (period of 20 ns), and desired width between the rising and corresponding falling edges of PPS1 signal output is 80 ns (that is, four units of sub second increment value), then you should program value 3 (4 − 1) in this register. ***Note)*** The value programmed in this register must be lesser than the value programmed in PPS1 Interval Register (PPS1_Interval).	R/W

6.4.98 Bus_Mode — Bus Mode Register

Address: GMAC1: 4400 1000h
GMAC2: 4400 3000h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	RIB	—	PRWG		TXPR	MB	AAL	PBLx8	USP	RPBL						FB
Value after reset	0	X	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PR		PBL						ATDS	DSL					DA	SWR
Value after reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1

Table 6.101 Bus_Mode Register Contents (1/2)

Bit Position	Bit Name	Function	R/W
b31	RIB	Rebuild INCRx Burst This bit is reserved and read-only (RO).	R
b30	Reserved	Reserved	R
b29, b28	PRWG	Channel Priority Weights This field is reserved and read-only (RO).	R
b27	TXPR	Transmit Priority This bit is reserved and read-only (RO).	R
b26	MB	Mixed Burst This bit is reserved and read-only (RO).	R
b25	AAL	Address Aligned Beats When this bit is set high and the FB bit is equal to 1, the AXI interface generates all bursts aligned to the start address LS bits. If the FB bit is equal to 0, the first burst (accessing the data buffer's start address) is not aligned, but subsequent bursts are aligned to the address.	R/W
b24	PBLx8	PBLx8 Mode When set high, this bit multiplies the programmed PBL value (Bits[22:17] and Bits[13:8]) eight times. Therefore, the DMA transfers the data in 8, 16 beats depending on the PBL value. The maximum number of beats to be transferred in one DMA transaction is 16 beats.	R/W
b23	USP	Use Separate PBL When set high, this bit configures the Rx DMA to use the value configured in Bits[22:17] as PBL. The PBL value in Bits[13:8] is applicable only to the Tx DMA operations. When reset to low, the PBL value in Bits[13:8] is applicable for both DMA engines.	R/W
b22 to b17	RPBL	Rx DMA PBL This field indicates the maximum number of beats to be transferred in one Rx DMA transaction. This is the maximum value that is used in a single block Read or Write. The Rx DMA always attempts to burst as specified in the RPBL bit each time it starts a Burst transfer on the host bus. You can program RPBL with values of 1, 2, 4, 8, and 16. Any other value results in undefined behavior. This field is valid and applicable only when USP is set high.	R/W
b16	FB	Fixed Burst This bit controls whether the AXI Master interface performs fixed burst transfers or not. When set, the AXI interface uses only SINGLE, INCR4, INCR8, or INCR16 during start of the normal burst transfers. When reset, the AXI interface uses SINGLE and INCR burst transfer operations. For more information, see Bit 0 (UNDEF) of the AXI Bus Mode register.	R/W
b15, b14	PR	Priority Ratio These bits are reserved and read-only (RO).	R

Table 6.101 Bus_Mode Register Contents (2/2)

Bit Position	Bit Name	Function	R/W
b13 to b8	PBL	Programmable Burst Length These bits indicate the maximum number of beats to be transferred in one DMA transaction. This is the maximum value that is used in a single block Read or Write. The DMA always attempts 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, and 16. Any other value results in undefined behavior. When USP is set high, this PBL value is applicable only for Tx DMA transactions.	R/W
b7	ATDS	Enhanced Descriptor Size When set, the size of the enhanced descriptor increases to 32 bytes (8 DWORDS). When reset, the descriptor size reverts back to 4 DWORDs (16 bytes).	R/W
b6 to b2	DSL	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 the DSL value is equal to zero, the descriptor table is taken as contiguous by the DMA in Ring mode.	R/W
b1	DA	DMA Arbitration Scheme This bit is reserved and read-only (RO).	R
b0	SWR	Software Reset When this bit is set, the MAC DMA Controller resets the logic and all internal registers of the GMAC. It is cleared automatically after the reset operation has completed in all of the GMAC clock domains. Before reprogramming any register of the GMAC, you should read a zero (0) value in this bit. ***Note*)** • The Software reset function is driven only by this bit. • The reset operation is completed only when all resets in all active clock domains are de-asserted. Therefore, it is essential that all the PHY inputs clocks (applicable for the selected PHY interface) are present for the software reset completion. • The time to complete the software reset operation depends on the frequency of the slowest active clock.	R/W

6.4.99 Transmit_Poll_Demand — Transmit Poll Demand Register

Address: GMAC1: 4400 1004h
GMAC2: 4400 3004h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TPD															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TPD															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6.102 Transmit_Poll_Demand Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	TPD	Transmit Poll Demand When these bits are written with any value, the DMA reads the current descriptor pointed to by Current Host Transmit Descriptor Register (Current_Host_Transmit_Descriptor). If that descriptor is not available (owned by the Host), the transmission returns to the Suspend state and the Bit 2 (TU) of Status Register (Status) is asserted. If the descriptor is available, the transmission resumes.	R/W

6.4.100 Receive_Poll_Demand — Receive Poll Demand Register

Address: GMAC1: 4400 1008h
GMAC2: 4400 3008h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	RPD															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RPD															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6.103 Receive_Poll_Demand Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	RPD	Receive Poll Demand When these bits are written with any value, the DMA reads the current descriptor pointed to by Current Host Receive Descriptor Register (Current_Host_Receive_Descriptor). If that descriptor is not available (owned by the Host), the reception returns to the Suspended state and the Bit 7 (RU) of Status Register (Status) is not asserted. If the descriptor is available, the Rx DMA returns to the active state.	R/W

6.4.101 Receive_Descriptor_List_Address — Receive Descriptor List Address Register

Address: GMAC1: 4400 100Ch
GMAC2: 4400 300Ch

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	RDESLA_32bit															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RDESLA_32bit														—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	X	X

Table 6.104 Receive_Descriptor_List_Address Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b2	RDESLA_32bit	Start of Receive List This field contains the base address of the first descriptor in the Receive Descriptor list.	R/W
b1, b0	Reserved	The LSB bits (1:0) for 32-bit bus width are ignored and internally taken as all-zero by the DMA. Therefore, these LSB bits are read-only (RO).	R

6.4.102 Transmit_Descriptor_List_Address — Transmit Descriptor List Address Register

Address: GMAC1: 4400 1010h
GMAC2: 4400 3010h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TDESLA_32bit															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TDESLA_32bit														—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	X	X

Table 6.105 Transmit_Descriptor_List_Address Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b2	TDESLA_32bit	Start of Transmit List This field contains the base address of the first descriptor in the Transmit Descriptor list.	R/W
b1, b0	Reserved	The LSB bits (1:0) for 32 bit bus width are ignored are internally taken as all zero by the DMA. Therefore, these LSB bits are read only (RO).	R

6.4.103 Status — Status Register

Address: GMAC1: 4400 1014h
GMAC2: 4400 3014h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	GLPII	TTI	GPI	GMI	—	EB			TS			RS			NIS
Value after reset	X	0	0	0	0	X	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	AIS	ERI	FBI	—	—	ETI	RWT	RPS	RU	RI	UNF	OVF	TJT	TU	TPS	TI
Value after reset	0	0	0	X	X	0	0	0	0	0	0	0	0	0	0	0

Table 6.106 Status Register Contents (1/3)

Bit Position	Bit Name	Function	R/W
b31	Reserved	Reserved	R
b30	GLPII	GMAC LPI Interrupt This bit indicates an interrupt event in the LPI logic of the GMAC. To reset this bit to 1’b0, the software must read the corresponding registers in the GMAC to get the exact cause of the interrupt and clear its source. When this bit is high, the interrupt signal from the GMAC (GMAC[m]_SBD_Int) is high.	R
b29	TTI	Timestamp Trigger Interrupt This bit indicates an interrupt event in the Timestamp Generator block of GMAC. The software must read the corresponding registers in the GMAC to get the exact cause of interrupt and clear its source to reset this bit to 1’b0. When this bit is high, the interrupt signal from the GMAC subsystem (GMAC[m]_SBD_Int) is high.	R
b28	GPI	GMAC PMT Interrupt This bit indicates an interrupt event in the PMT module of the GMAC. The software must read the PMT Control and Status Register in the GMAC to get the exact cause of interrupt and clear its source to reset this bit to 1’b0. The interrupt signal from the GMAC subsystem (GMAC[m]_SBD_Int) is high when this bit is high. ***Note)*** The GPI and GMAC[m]_PMT_Int interrupts are generated in different clock domains.	R
b27	GMI	GMAC MMC Interrupt This bit reflects an interrupt event in the MMC module of the GMAC. The software must read the corresponding registers in the GMAC to get the exact cause of interrupt and clear the source of interrupt to make this bit as 1’b0. The interrupt signal from the GMAC subsystem (GMAC[m]_SBD_Int) is high when this bit is high.	R
b26	Reserved	Reserved	R
b25 to b23	EB	Error Bits This field indicates the type of error that caused a Bus Error, for example, error response on the AXI interface. This field is valid only when Bit 13 (FBI) is set. This field does not generate an interrupt. 3’b000: Error during Rx DMA Write Data Transfer 3’b001 and 3’b010: Reserved 3’b011: Error during Tx DMA Read Data Transfer 3’b100: Error during Rx DMA Descriptor Write Access 3’b101: Error during Tx DMA Descriptor Write Access 3’b110: Error during Rx DMA Descriptor Read Access 3’b111: Error during Tx DMA Descriptor Read Access	R

Table 6.106 Status Register Contents (2/3)

Bit Position	Bit Name	Function	R/W
b22 to b20	TS	Transmit Process State This field indicates the Transmit DMA FSM state. This field does not generate an interrupt. 3'b000: Stopped; Reset or Stop Transmit Command issued 3'b001: Running; Fetching Transmit Transfer Descriptor 3'b010: Running; Waiting for status 3'b011: Running; Reading Data from host memory buffer and queuing it to transmit buffer (Tx FIFO) 3'b100: TIME_STAMP write state 3'b101: Reserved for future use 3'b110: Suspended; Transmit Descriptor Unavailable or Transmit Buffer Underflow 3'b111: Running; Closing Transmit Descriptor	R
b19 to b17	RS	Received Process State This field indicates the Receive DMA FSM state. This field does not generate an interrupt. 3'b000: Stopped: Reset or Stop Receive Command issued 3'b001: Running: Fetching Receive Transfer Descriptor 3'b010: Reserved for future use 3'b011: Running: Waiting for receive packet 3'b100: Suspended: Receive Descriptor Unavailable 3'b101: Running: Closing Receive Descriptor 3'b110: TIME_STAMP write state 3'b111: Running: Transferring the receive packet data from receive buffer to host memory	R
b16	NIS	Normal Interrupt Summary Normal Interrupt Summary bit value is the logical OR of the following bits when the corresponding interrupt bits are enabled in Interrupt Enable Register (Interrupt_Enable): ● Interrupt_Enable [0]: Transmit Interrupt ● Interrupt_Enable [2]: Transmit Buffer Unavailable ● Interrupt_Enable [6]: Receive Interrupt ● Interrupt_Enable [14]: Early Receive Interrupt Only unmasked bits (interrupts for which interrupt enable is set in Interrupt Enable Register (Interrupt_Enable)) 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, which causes NIS to be set, is cleared.	R/W
b15	AIS	Abnormal Interrupt Summary Abnormal Interrupt Summary bit value is the logical OR of the following bits when the corresponding interrupt bits are enabled in Interrupt Enable Register (Interrupt_Enable): ● Interrupt_Enable [1]: Transmit Process Stopped ● Interrupt_Enable [3]: Transmit Jabber Timeout ● Interrupt_Enable [4]: Receive FIFO Overflow ● Interrupt_Enable [5]: Transmit Underflow ● Interrupt_Enable [7]: Receive Buffer Unavailable ● Interrupt_Enable [8]: Receive Process Stopped ● Interrupt_Enable [9]: Receive Watchdog Timeout ● Interrupt_Enable [10]: Early Transmit Interrupt ● Interrupt_Enable [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, which causes AIS to be set, is cleared.	R/W
b14	ERI	Early Receive Interrupt This bit indicates that the DMA filled the first data buffer of the packet. This bit is cleared when the software writes 1 to this bit or Bit 6 (RI) of this register is set (whichever occurs earlier).	R/W

Table 6.106 Status Register Contents (3/3)

Bit Position	Bit Name	Function	R/W
b13	FBI	Fatal Bus Error Interrupt This bit indicates that a bus error occurred, as described in Bits[25:23]. When this bit is set, the corresponding DMA engine disables all of its bus accesses.	R/W
b12, b11	Reserved	Reserved	R
b10	ETI	Early Transmit Interrupt This bit indicates that the frame to be transmitted is fully transferred to the MTL Transmit FIFO.	R/W
b9	RWT	Receive Watchdog Timeout When set, this bit indicates that the Receive Watchdog Timer expired while receiving the current frame and the current frame is truncated after the watchdog timeout.	R/W
b8	RPS	Receive Process Stopped This bit is asserted when the Receive Process enters the Stopped state.	R/W
b7	RU	Receive Buffer Unavailable This bit indicates that the host owns the Next Descriptor in the Receive List and the DMA cannot acquire it. The 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, the Receive Process resumes when the next recognized incoming frame is received. This bit is set only when the previous Receive Descriptor is owned by the DMA.	R/W
b6	RI	Receive Interrupt This bit indicates that the frame reception is complete. When reception is complete, the Bit 31 of RDES1 (Disable Interrupt on Completion) is reset in the last Descriptor, and the specific frame status information is updated in the descriptor. The reception remains in the Running state.	R/W
b5	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.	R/W
b4	OVF	Receive Overflow This bit indicates that the Receive Buffer had an Overflow during frame reception. If the partial frame is transferred to the application, the overflow status is set in RDES0[11].	R/W
b3	TJT	Transmit Jabber Timeout This bit indicates that the Transmit Jabber Timer expired, which happens when the frame size exceeds 2,048 (10,240 bytes when the Jumbo frame is enabled). When the Jabber Timeout occurs, the transmission process is aborted and placed in the Stopped state. This causes the Transmit Jabber Timeout TDES0[14] flag to assert.	R/W
b2	TU	Transmit Buffer Unavailable This bit indicates that the host owns the Next Descriptor in the Transmit List and the DMA cannot acquire it. Transmission is suspended. Bits[22:20] explain the Transmit Process state transitions. To resume processing Transmit descriptors, the host should change the ownership of the descriptor by setting TDES0[31] and then issue a Transmit Poll Demand command.	R/W
b1	TPS	Transmit Process Stopped This bit is set when the transmission is stopped.	R/W
b0	TI	Transmit Interrupt This bit indicates that the frame transmission is complete. When transmission is complete, Bit 31 (OWN) of TDES0 is reset, and the specific frame status information is updated in the descriptor.	R/W

6.4.104 Operation_Mode — Operation Mode Register

Address: GMAC1: 4400 1018h
GMAC2: 4400 3018h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	DT	RSF	—	—	—	TSF	FTF	—	—	—	TTC
Value after reset	X	X	X	X	X	0	0	X	X	0	0	0	X	X	X	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TTC		ST	RFD		RFA		EFC	FEF	FUF	DGF	RTC		OSF	SR	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	X

Table 6.107 Operation_Mode Register Contents (1/3)

Bit Position	Bit Name	Function	R/W
b31 to b27	Reserved	Reserved	R
b26	DT	Disable Dropping of TCP/IP Checksum Error Frames When this bit is set, the GMAC does not drop the frames which only have 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 only in the encapsulated payload. When this bit is reset, all error frames are dropped if the FEF bit is reset.	R/W
b25	RSF	Receive Store and Forward When this bit is set, the MTL reads a frame from the Rx FIFO only after the complete frame has been written to it, ignoring the RTC bits. When this bit is reset, the Rx FIFO operates in the cut-through mode, subject to the threshold specified by the RTC bits.	R/W
b24 to b22	Reserved	Reserved	R/W
b21	TSF	Transmit Store and Forward When this bit is set, transmission starts when a full frame resides in the MTL Transmit FIFO. When this bit is set, the TTC values specified in Bits[16:14] are ignored. This bit should be changed only when the transmission is stopped.	R/W
b20	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 Tx FIFO is lost or flushed. This bit is cleared internally when the flushing operation is completed. The Operation Mode register should not be written to until this bit is cleared. The data which is already accepted by the GMAC transmitter is not flushed. It is scheduled for transmission and results in underflow and runt frame transmission. ***Note*)** The flush operation is complete only when the Tx FIFO is emptied of its contents and all the pending Transmit Status of the transmitted frames are accepted by the host. To complete this flush operation, the PHY transmit clock is required to be active.	R/W
b19 to b17	Reserved	Reserved	R

Table 6.107 Operation_Mode Register Contents (2/3)

Bit Position	Bit Name	Function	R/W
b16 to b14	TTC	Transmit Threshold Control These bits control the threshold level of the MTL Transmit FIFO. Transmission 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 Bit 21 (TSF) is reset. 3'b000: 64 3'b001: 128 3'b010: 192 3'b011: 256 3'b100: 40 3'b101: 32 3'b110: 24 3'b111: 16	R/W
b13	ST	Start or Stop Transmission Command When this bit is set, transmission 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 Transmit Descriptor List Address Register (Transmit_Descriptor_List_Address), or from the position retained when transmission was stopped previously. If the DMA does not own the current descriptor, transmission enters the Suspended state and Bit 2 (Transmit Buffer Unavailable) of Status Register (Status) is set. The Start Transmission command is effective only when transmission is stopped. If the command is issued before setting Transmit Descriptor List Address Register (Transmit_Descriptor_List_Address), 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 it becomes the current position when transmission is restarted. To change the list address, you need to program Transmit Descriptor List Address Register (Transmit_Descriptor_List_Address) with a new value when this bit is reset. The new value is considered when this bit is set again. The stop transmission command is effective only when the transmission of the current frame is complete or the transmission is in the Suspended state.	R/W
b12, b11	RFD	Threshold for Deactivating Flow Control (in half-duplex and full-duplex) These bits control the threshold (Fill-level of Rx FIFO) at which the flow control is de-asserted after activation. 2'b00: Full minus 1 KB, that is, FULL - 1 KB 2'b01: Full minus 2 KB, that is, FULL - 2 KB 2'b10: Full minus 3 KB, that is, FULL - 3 KB 2'b11: Full minus 4 KB, that is, FULL - 4 KB The de-assertion is effective only after flow control is asserted. ***Note*)** For proper flow control, the value programmed in the RFD field should be equal to or more than the value programmed in the RFA field.	R/W
b10, b9	RFA	Threshold for Activating Flow Control (in half-duplex and full-duplex) These bits control the threshold (Fill level of Rx FIFO) at which the flow control is activated. 2'b00: Full minus 1 KB, that is, FULL - 1 KB 2'b01: Full minus 2 KB, that is, FULL - 2 KB 2'b10: Full minus 3 KB, that is, FULL - 3 KB 2'b11: Reserved These values are applicable only when Bit 8 (EFC) is set high.	R/W
b8	EFC	Enable HW Flow Control When this bit is set, the flow control signal operation based on the fill-level of Rx FIFO is enabled. When reset, the flow control operation is disabled.	R/W

Table 6.107 Operation_Mode Register Contents (3/3)

Bit Position	Bit Name	Function	R/W
b7	FEF	Forward Error Frames When this bit is reset, the Rx FIFO drops frames with error status (CRC error, collision error, GMII_RXER, giant frame, watchdog timeout, or overflow). However, if the start byte (write) pointer of a frame is already transferred to the read controller side (in Threshold mode), then the frame is not dropped. When the FEF bit is set, all frames except runt error frames are forwarded to the DMA. If the Bit 25 (RSF) is set and the Rx FIFO overflows when a partial frame is written, then the frame is dropped irrespective of the FEF bit setting. However, if the Bit 25 (RSF) is reset and the Rx FIFO overflows when a partial frame is written, then a partial frame may be forwarded to the DMA.	R/W
b6	FUF	Forward Undersized Good Frames When set, the Rx FIFO forwards Undersized frames (frames with no Error and length less than 64 bytes) including pad-bytes and CRC. When reset, the Rx FIFO drops all frames of less than 64 bytes, unless a frame is already transferred because of the lower value of Receive Threshold, for example, RTC = 2’b01.	R/W
b5	DGF	Drop Giant Frames When set, the GMAC drops the received giant frames in the Rx FIFO, that is, frames that are larger than the computed giant frame limit. When reset, the GMAC does not drop the giant frames in the Rx FIFO.	R/W
b4, b3	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 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. 2’b00: 64 2’b01: 32 2’b10: 96 2’b11: 128	R/W
b2	OSF	Operate on Second Frame When this bit is set, it instructs the DMA to process the second frame of the Transmit data even before the status for the first frame is obtained.	R/W
b1	SR	Start or Stop Receive When this bit is set, the Receive process is placed in the Running state. The DMA attempts to acquire the descriptor from the Receive list and processes the incoming frames. The descriptor acquisition is attempted from the current position in the list, which is the address set by Receive Descriptor List Address Register (Receive_Descriptor_List_Address) or the position retained when the Receive process was previously stopped. If the DMA does not own the descriptor, reception is suspended and Bit 7 (Receive Buffer Unavailable) of Status Register (Status) is set. The Start Receive command is effective only when the reception has stopped. If the command is issued before setting Receive Descriptor List Address Register (Receive_Descriptor_List_Address), the DMA behavior is unpredictable. When this bit is cleared, the Rx DMA operation is stopped after the transfer of the current frame. The next descriptor position in the Receive 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.	R/W
b0	Reserved	Reserved	R

6.4.105 Interrupt_Enable — Interrupt Enable Register

Address: GMAC1: 4400 101Ch
GMAC2: 4400 301Ch

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	NIE
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	AIE	ERE	FBE	—	—	ETE	RWE	RSE	RUE	RIE	UNE	OVE	TJE	TUE	TSE	TIE
Value after reset	0	0	0	X	X	0	0	0	0	0	0	0	0	0	0	0

Table 6.108 Interrupt_Enable Register Contents (1/2)

Bit Position	Bit Name	Function	R/W
b31 to b17	Reserved	Reserved	R
b16	NIE	Normal Interrupt Summary Enable When this bit is set, normal interrupt summary is enabled. When this bit is reset, normal interrupt summary is disabled. This bit enables the following interrupts in Status Register): ● Status [0]: Transmit Interrupt ● Status [2]: Transmit Buffer Unavailable ● Status [6]: Receive Interrupt ● Status [14]: Early Receive Interrupt	R/W
b15	AIE	Abnormal Interrupt Summary Enable When this bit is set, abnormal interrupt summary is enabled. When this bit is reset, the abnormal interrupt summary is disabled. This bit enables the following interrupts in Status Register: ● Status [1]: Transmit Process Stopped ● Status [3]: Transmit Jabber Timeout ● Status [4]: Receive Overflow ● Status [5]: Transmit Underflow ● Status [7]: Receive Buffer Unavailable ● Status [8]: Receive Process Stopped ● Status [9]: Receive Watchdog Timeout ● Status [10]: Early Transmit Interrupt ● Status [13]: Fatal Bus Error	R/W
b14	ERE	Early Receive Interrupt Enable When this bit is set with Normal Interrupt Summary Enable (Bit 16), the Early Receive Interrupt is enabled. When this bit is reset, the Early Receive Interrupt is disabled.	R/W
b13	FBE	Fatal Bus Error Enable When this bit is set with Abnormal Interrupt Summary Enable (Bit 15), the Fatal Bus Error Interrupt is enabled. When this bit is reset, the Fatal Bus Error Enable Interrupt is disabled.	R/W
b12, b11	Reserved	Reserved	R
b10	ETE	Early Transmit Interrupt Enable When this bit is set with an Abnormal Interrupt Summary Enable (Bit 15), the Early Transmit Interrupt is enabled. When this bit is reset, the Early Transmit Interrupt is disabled.	R/W
b9	RWE	Receive Watchdog Timeout Enable When this bit is set with Abnormal Interrupt Summary Enable (Bit 15), the Receive Watchdog Timeout Interrupt is enabled. When this bit is reset, the Receive Watchdog Timeout Interrupt is disabled.	R/W

Table 6.108 Interrupt_Enable Register Contents (2/2)

Bit Position	Bit Name	Function	R/W
b8	RSE	Receive Stopped Enable When this bit is set with Abnormal Interrupt Summary Enable (Bit 15), the Receive Stopped Interrupt is enabled. When this bit is reset, the Receive Stopped Interrupt is disabled.	R/W
b7	RUE	Receive Buffer Unavailable Enable When this bit is set with Abnormal Interrupt Summary Enable (Bit 15), the Receive Buffer Unavailable Interrupt is enabled. When this bit is reset, the Receive Buffer Unavailable Interrupt is disabled.	R/W
b6	RIE	Receive Interrupt Enable When this bit is set with Normal Interrupt Summary Enable (Bit 16), the Receive Interrupt is enabled. When this bit is reset, the Receive Interrupt is disabled.	R/W
b5	UNE	Underflow Interrupt Enable When this bit is set with Abnormal Interrupt Summary Enable (Bit 15), the Transmit Underflow Interrupt is enabled. When this bit is reset, the Underflow Interrupt is disabled.	R/W
b4	OVE	Overflow Interrupt Enable When this bit is set with Abnormal Interrupt Summary Enable (Bit 15), the Receive Overflow Interrupt is enabled. When this bit is reset, the Overflow Interrupt is disabled.	R/W
b3	TJE	Transmit Jabber Timeout Enable When this bit is set with Abnormal Interrupt Summary Enable (Bit 15), the Transmit Jabber Timeout Interrupt is enabled. When this bit is reset, the Transmit Jabber Timeout Interrupt is disabled.	R/W
b2	TUE	Transmit Buffer Unavailable Enable When this bit is set with Normal Interrupt Summary Enable (Bit 16), the Transmit Buffer Unavailable Interrupt is enabled. When this bit is reset, the Transmit Buffer Unavailable Interrupt is disabled.	R/W
b1	TSE	Transmit Stopped Enable When this bit is set with Abnormal Interrupt Summary Enable (Bit 15), the Transmission Stopped Interrupt is enabled. When this bit is reset, the Transmission Stopped Interrupt is disabled.	R/W
b0	TIE	Transmit Interrupt Enable When this bit is set with Normal Interrupt Summary Enable (Bit 16), the Transmit Interrupt is enabled. When this bit is reset, the Transmit Interrupt is disabled.	R/W

6.4.106 Missed_Frame_And_Buffer_Overflow_Counter — Missed Frame and Buffer Overflow Counter Register

Address: GMAC1: 4400 1020h
GMAC2: 4400 3020h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	OVFCNTOVF	OVFFRMCNT											MISCNTOVF
Value after reset	X	X	X	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MISFRMCNT															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6.109 Missed_Frame_And_Buffer_Overflow_Counter Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b29	Reserved	Reserved	R
b28	OVFCNTOVF	Overflow Bit for FIFO Overflow Counter This bit is set every time the Overflow Frame Counter (Bits[27:17]) overflows, that is, the Rx FIFO overflows with the overflow frame counter at maximum value. In such a scenario, the overflow frame counter is reset to all-zeros and this bit indicates that the rollover happened.	R
b27 to b17	OVFFRMCNT	Overflow Frame Counter This field indicates the number of frames missed by the application. This counter is incremented each time the MTL FIFO overflows. The counter is cleared when this bit is read.	R
b16	MISCNTOVF	Overflow Bit for Missed Frame Counter This bit is set every time Missed Frame Counter (Bits[15:0]) overflows, that is, the DMA discards an incoming frame because of the Host Receive Buffer being unavailable with the missed frame counter at maximum value. In such a scenario, the Missed frame counter is reset to all-zeros and this bit indicates that the rollover happened.	R
b15 to b0	MISFRMCNT	Missed Frame Counter This field indicates the number of frames missed by the controller because of the Host Receive Buffer being unavailable. This counter is incremented each time the DMA discards an incoming frame. The counter is cleared when this bit is read.	R

6.4.107 Receive_Interrupt_Watchdog_Timer — Receive Interrupt Watchdog Timer Register

Address: GMAC1: 4400 1024h
GMAC2: 4400 3024h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	RIWT							
Value after reset	X	X	X	X	X	X	X	X	0	0	0	0	0	0	0	0

Table 6.110 Receive_Interrupt_Watchdog_Timer Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b8	Reserved	Reserved	R
b7 to b0	RIWT	RI Watchdog Timer Count This bit indicates the number of system clock cycles multiplied by 256 for which the watchdog timer is set. The watchdog timer gets triggered with the programmed value after the Rx DMA completes the transfer of a frame for which the RI status bit is not set because of the setting in the corresponding 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 the RI bit is set high because of automatic setting of RI as per RDES1[31] of any received frame.	R/W

6.4.108 AXI_Bus_Mode — AXI Bus Mode Register

Address: GMAC1: 4400 1028h
GMAC2: 4400 3028h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	EN_LPI	LPI_XIT_FRM	—	—	—	—	—	—	—	—	WR_OSR_LMT		—	—	RD_OSR_LMT	
Value after reset	0	0	X	X	X	X	X	X	X	X	0	1	X	X	0	1

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	ONEKBBE	AXI_AAL	—	—	—	—	—	—	—	—	BLEN16	BLEN8	BLEN4	UNDEF
Value after reset	X	X	0	0	X	X	X	X	X	X	X	X	0	0	0	1

Table 6.111 AXI_Bus_Mode Register Contents (1/2)

Bit Position	Bit Name	Function	R/W
b31	EN_LPI	Enable Low Power Interface (LPI) When set to 1, this bit enables the LPI mode and accepts the LPI request from the AXI System Clock controller. When set to 0, this bit disables the LPI mode and always denies the LPI request from the AXI System Clock controller.	R/W
b30	LPI_XIT_FRM	Unlock on Magic Packet or Remote Wake-Up Frame When set to 1, this bit enables the GMAC-AXI to come out of the LPI mode only when the Magic Packet or Remote Wake-Up Packet is received. When set to 0, this bit enables the GMAC-AXI to come out of LPI mode when any frame is received.	R/W
b29 to b22	Reserved	Reserved	R
b21, b20	WR_OSR_LMT	AXI Maximum Write OutStanding Request Limit This value limits the maximum outstanding request on the AXI write interface. Maximum outstanding requests = WR_OSR_LMT+1	R/W
b19, b18	Reserved	Reserved	R
b17, b16	RD_OSR_LMT	AXI Maximum Read OutStanding Request Limit This value limits the maximum outstanding request on the AXI read interface. Maximum outstanding requests = RD_OSR_LMT+1	R/W
b15, b14	Reserved	Reserved	R
b13	ONEKBBE	1 KB Boundary Crossing Enable for the GMAC-AXI Master When set, the GMAC-AXI Master performs burst transfers that do not cross 1 KB boundary. When reset, the GMAC-AXI Master performs burst transfers that do not cross 4 KB boundary.	R/W
b12	AXI_AAL	Address-Aligned Beats This bit is read-only bit and reflects the Bit 25 (AAL) of Bus Mode Register (Bus_Mode). When this bit is set to 1, the GMAC-AXI performs address-aligned burst transfers on both read and write channels.	R
b11 to b4	Reserved	Reserved	R
b3	BLEN16	AXI Burst Length 16 When this bit is set to 1 or UNDEF is set to 1, the GMAC-AXI is allowed to select a burst length of 16 on the AXI Master interface.	R/W
b2	BLEN8	AXI Burst Length 8 When this bit is set to 1, the GMAC-AXI is allowed to select a burst length of 8 on the AXI Master interface. Setting this bit has no effect when UNDEF is set to 1.	R/W

Table 6.111 AXI_Bus_Mode Register Contents (2/2)

Bit Position	Bit Name	Function	R/W
b1	BLEN4	AXI Burst Length 4 When this bit is set to 1, the GMAC-AXI is allowed to select a burst length of 4 on the AXI Master interface. Setting this bit has no effect when UNDEF is set to 1.	R/W
b0	UNDEF	AXI Undefined Burst Length This bit is read-only bit and indicates the complement (invert) value of Bit 16 (FB) in Bus Mode Register (Bus_Mode[16]). ● When this bit is set to 1, the GMAC-AXI is allowed to perform any burst length equal to or below the maximum allowed burst length programmed in Bits[3]. ● When this bit is set to 0, the GMAC-AXI is allowed to perform only fixed burst lengths as indicated by BLEN16, BLEN8, or BLEN4, or a burst length of 1.	R

6.4.109 AXI_Status — AXI Status Register

Address: GMAC1: 4400 102Ch
GMAC2: 4400 302Ch

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	AXIRDSTS	AXWHSTS
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	0	0

Table 6.112 AXI_Status Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b2	Reserved	Reserved	R
b1	AXIRDSTS	AXI Master Read Channel Status When high, it indicates that AXI Master's read channel is active and transferring data.	R
b0	AXWHSTS	AXI Master Write Channel When high, it indicates that AXI Master's write channel is active and transferring data.	R

6.4.110 Current_Host_Transmit_Descriptor — Current Host Transmit Descriptor Register

Address: GMAC1: 4400 1048h
GMAC2: 4400 3048h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	CURTDESAPTR															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CURTDESAPTR															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6.113 Current_Host_Transmit_Descriptor Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	CURTDESAPTR	Host Transmit Descriptor Address Pointer Cleared on Reset. Pointer updated by the DMA during operation.	R

6.4.111 Current_Host_Receive_Descriptor — Current Host Receive Descriptor Register

Address: GMAC1: 4400 104Ch
GMAC2: 4400 304Ch

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	CURRDESAPTR															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CURRDESAPTR															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6.114 Current_Host_Receive_Descriptor Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	CURRDESAPTR	Host Receive Descriptor Address Pointer Cleared on Reset. Pointer updated by the DMA during operation.	R

6.4.112 Current_Host_Transmit_Buffer_Address — Current Host Transmit Buffer Address Register

Address: GMAC1: 4400 1050h
GMAC2: 4400 3050h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	CURTBUFAPTR															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CURTBUFAPTR															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6.115 Current_Host_Transmit_Buffer_Address Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	CURTBUFAPTR	Host Transmit Buffer Address Pointer Cleared on Reset. Pointer updated by the DMA during operation.	R

6.4.113 Current_Host_Receive_Buffer_Address — Current Host Receive Buffer Address Register

Address: GMAC1: 4400 1054h
GMAC2: 4400 3054h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	CURRBUFAPTR															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CURRBUFAPTR															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6.116 Current_Host_Receive_Buffer_Address Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	CURRBUFAPTR	Host Receive Buffer Address Pointer Cleared on Reset. Pointer updated by the DMA during operation.	R

6.4.114 HW_Feature — HW Feature Register

Address: GMAC1: 4400 1058h
GMAC2: 4400 3058h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	ACTPHYIF			SAVLANINS	FLEXIPPSEN	INTTSEN	ENHDESSEL	TXCHCNT		RXCHCNT		RXFIFOSIZE	RXTYP2COE	RXTYP1COE	TXCOESEL
Value after reset	X	0	0	0	0	1/0	1	1	0	0	0	0	1	1	0	1

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	AVSEL	EEESEL	TSVER2SEL	TSVER1SEL	MMCSEL	MGKSEL	RWKSEL	SMASEL	L3L4FLTREN	—	ADDMACADRSEL	HASHSEL	EXTHASHEN	HDSEL	GMIISEL	MIISEL
Value after reset	0	1	1	0	1	1	1	1	0	0	1	1	1	1	1	1

Table 6.117 HW_Feature Register Contents

Bit Position	Bit Name	Function	R/W
b31	Reserved	Reserved	R
b30 to b28	ACTPHYIF	Active or Selected PHY interface 3'b000: GMII or MII All Others: Reserved	R
b27	SAVLANINS	Source Address or VLAN Insertion	R
b26	FLEXIPPSEN	Flexible Pulse-Per-Second Output (GMAC1: 1, GMAC2: 0)	R
b25	INTTSEN	Timestamping with Internal System Time	R
b24	ENHDESSEL	Enhanced Descriptor	R
b23, b22	TXCHCNT	Number of additional Tx channels	R
b21, b20	RXCHCNT	Number of additional Rx channels	R
b19	RXFIFOSIZE	Rx FIFO > 2,048 Bytes	R
b18	RXTYP2COE	IP Checksum Offload (Type 2) in Rx	R
b17	RXTYP1COE	IP Checksum Offload (Type 1) in Rx	R
b16	TXCOESEL	Checksum Offload in Tx	R
b15	AVSEL	AV Feature	R
b14	EEESEL	Energy Efficient Ethernet	R
b13	TSVER2SEL	IEEE 1588-2008 Advanced Timestamp	R
b12	TSVER1SEL	Only IEEE 1588-2002 Timestamp	R
b11	MMCSEL	RMON Module	R
b10	MGKSEL	PMT Magic Packet	R
b9	RWKSEL	PMT Remote wake-up	R
b8	SMASEL	SMA (MDIO) Interface	R
b7	L3L4FLTREN	Layer 3 and Layer 4 Filter Feature	R
b6	Reserved	Reserved	R
b5	ADDMACADRSEL	Multiple MAC Address Registers	R
b4	HASHSEL	HASH Filter	R
b3	EXTHASHEN	Expanded DA Hash Filter	R
b2	HDSEL	Half-Duplex support	R
b1	GMIISEL	1000 Mbps support	R
b0	MIISEL	10 or 100 Mbps support	R

6.5 Operation

This chapter provides the instructions for initializing the DMA or MAC registers in the proper sequence.

6.5.1 Initializing

The initializing sequence in this section is a example for preparation of system environments for using GMAC under configuration below.

GMAC configuration of this example:

- GMAC1 is connected to external port1 through RGMII/RMII Converter.
- GMAC2 is connected to external port2 through RGMII/RMII Converter.

“Initializing DMA” and “Initializing GMAC” sequence is followed after the sequence of this section.

6.5.1.1 Initializing Operation

For initializing operation, complete the following flowchart:

Note 1. RZ/N1 User's Manual: System Introduction, Multiplexing, Electrical and Mechanical Information

Note 2. Ethernet MAC 10/100/1000 (GMAC): Register Descriptions, IEEE 1588 Timestamp Registers

Figure 6.2 Initializing Flowchart

6.5.2 Initializing DMA

The initialization sequence in this section is used for GMAC configurations

Complete the following steps to initialize the DMA:

1. Provide a software reset. This resets all of the GMAC internal registers and logic. (Bus Mode Register (Bus_Mode) - bit 0).
2. Wait for the completion of the reset process (poll bit 0 of the Bus Mode Register (Bus_Mode), which is only cleared after the reset operation is completed).
3. Poll the bits of AXI Status Register (AXI_Status) to confirm that all previously initiated (before soft reset) or ongoing AXI transactions are complete.

CAUTION

If the application cannot poll the AXI Status Register (AXI_Status) after soft reset (because of performance reasons), then it is recommended that you continue with the next steps and check this register again (as mentioned in Step 11) before triggering the DMA operations.

4. Program the following fields to initialize the Bus Mode Register by setting values in Bus Mode Register:
   - AAL
   - Fixed burst or undefined burst
   - Burst length values and burst mode values.
   - Descriptor Length
5. Program the AXI Interface options in AXI Bus Mode Register (AXI_Bus_Mode).
6. Create a proper descriptor chain for transmit and receive. In addition, ensure that the receive descriptors are owned by DMA (bit 31 of descriptor 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 chain before reusing any of the descriptors.
8. Initialize receive and transmit descriptor list address with the base address of the transmit and receive descriptor (Receive Descriptor List Address Register (Receive_Descriptor_List_Address) and Transmit Descriptor List Address Register (Transmit_Descriptor_List_Address) respectively).
9. Program the following fields to initialize the mode of operation in Operation Mode Register (Operation_Mode)
   - Receive and Transmit Store And Forward
   - Receive and Transmit Threshold Control (RTC and TTC)
   - Hardware Flow Control enable
   - Flow Control Activation and De-activation thresholds for MTL Receive and Transmit FIFO (RFA and RFD)
   - Error Frame and undersized good frame forwarding enable
   - OSF Mode

10. Enable the interrupts by programming Interrupt Enable Register (Interrupt_Enable).

CAUTION

Perform Step 11 only if you did not perform Step 3.

11. Read AXI Status Register (AXI_Status) to confirm that all previous AXI transactions are complete.

CAUTION

If any previous transaction is still in progress when you read the AXI Status Register, it is strongly recommended to check the Slave components addressed by the AXI master interface of GMAC.

12. Start the Receive and Transmit DMA by setting SR (bit 1) and ST (bit 13) of the Operation Mode Register (Operation_Mode).

6.5.3 Initializing GMAC

The following GMAC Initialization operations can be performed after DMA initialization. If the GMAC initialization is done before the DMA is set up, then enable the GMAC receiver (last step in following sequence) only after the DMA is active. Otherwise, received frames fills the Rx FIFO and overflow.

1. Program the GMII Address Register (GMII_Address) for controlling the management cycles for external PHY. For example, Physical Layer Address PA (bits 15-11). In addition, set bit 0 (GMII Busy) for writing into PHY and reading from PHY.
2. Read the 16-bit data of GMII Data Register (GMII_Data) from the PHY for link up, speed of operation, and mode of operation, by specifying the appropriate address value in bits 15-11 of GMII Address Register (GMII_Address).
3. Provide the MAC address MAC Address0 High Register (MAC_Address0_High) and MAC Address0 Low Register (MAC_Address0_Low).
4. Program the following fields to set the appropriate filters for the incoming frames in MAC Frame Filter Register:
   - Receive All
   - Promiscuous mode
   - Hash or Perfect Filter
   - Unicast, multicast, broadcast, and control frames filter settings
5. Program the following fields for proper flow control in Flow Control Register (Flow_Control):
   - Pause time and other Pause frame control bits
   - Receive and Transmit Flow control bits
   - Flow Control Busy/Backpressure Activate
6. Program the Interrupt Mask register bits, as required.
7. Program the appropriate fields in MAC Configuration Register (MAC_Configuration). For example, Inter frame gap while transmission and jabber disable. Based on the Auto negotiation you can set the Duplex mode (bit 11) or port select (bit 15).
8. Set Bit 3 (TE) and Bit 2 (RE) in MAC Configuration Register (MAC_Configuration).

CAUTION

Do not change the configuration (such as duplex mode, speed, port, or loopback) when the GMAC is actively transmitting or receiving. The Software should change these parameters only when the GMAC transmitter and receiver are not active.

6.5.4 Performing Normal Receive and Transmit Operation

For normal operation, complete the following steps:

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 are not owned by the DMA (or no descriptor is available), the DMA goes 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 Transmit Poll Demand Register (Transmit_Poll_Demand) and Receive Poll Demand Register (Receive_Poll_Demand).
4. The values of the current host transmitter or receiver descriptor address pointer can be read for the debug process (Current Host Transmit Descriptor Register (Current_Host_transmit_Descriptor) and Current Host Receive Descriptor Register (Current_Host_Receive_Descriptor)).
5. The values of the current host transmit buffer address pointer and receive buffer address pointer can be read for the debug process (Current Host Transmit Buffer Address Register (Current_Host_Transmit_Buffer_Address) and Current Host Receive Buffer Address Register (Current_Host_Receive_Buffer_Address)).

6.5.5 Stopping and Starting Transmission

Complete the following steps to pause the transmission for some time:

1. Disable the Transmit DMA (if applicable) by clearing Bit 13 (ST) of Operation Mode Register (Operation_Mode).
2. Wait for any previous frame transmissions to complete. You can check this by reading the appropriate bits of Debug Register (Debug).
3. Disable the GMAC transmitter and GMAC receiver by clearing Bit 3 (TE) and Bit 2 (RE) in MAC Configuration Register (MAC_Configuration).
4. Disable the Receive DMA (if applicable), after making sure that the data in the Rx FIFO is transferred to the system memory (by reading Debug Register (Debug)).
5. Make sure that both Tx FIFO and Rx FIFO are empty.
6. To restart the operation, first start the DMAs, and then enable the GMAC Transmitter and Receiver.

6.5.6 Programming Guidelines for GMII Link Transitions

6.5.6.1 Transmit and Receive Clocks are Running when the Link is Down

Complete the following steps when the link is down but the Transmit and Receive clocks are running:

1. Disable the Transmit DMA (if applicable), by clearing bit 13 (ST) of Operation Mode Register (Operation_Mode).
2. Disable the GMAC receiver by clearing Bit 2 (RE) of MAC Configuration Register (MAC_Configuration).
3. Wait for any previous frame transmissions to complete from the Tx FIFO. You can do this by reading the appropriate bits of Debug Register (Debug).
   -or-
   Flush the Tx FIFO for faster empty operation.
4. Disable the GMAC transmitter by clearing bit 3 (TE) in MAC Configuration Register (MAC_Configuration).
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 Tx DMA, and then enabling the GMAC Transmitter and Receiver

You do not need to disable the Rx DMA. As the Receiver is disabled, the FIFO does not get any data in the Rx FIFO.

6.5.6.2 Transmit and Receive Clocks are Stopped when the Link is Down

Complete the following steps when the link is down and the Transmit and Receive clocks are stopped:

1. Wait till the link is up and the Transmit and Receive clocks are active.
   When the Transmit and Receive clocks are stopped, then disabling the transmit or receive operations does not have any effect. Therefore, the software must wait till the link is up again.
2. Disable the Transmit DMA (if applicable), by clearing Bit 13 (ST) of the Operation Mode Register (Operation_Mode)
3. Disable the GMAC receiver by clearing Bit 2 (RE) of MAC Configuration Register (MAC_Configuration).
4. Wait for any previous frame transmissions to complete from the Tx FIFO. You can do this by reading the appropriate bits of Debug Register (Debug).
   -or-
   Flush the Tx FIFO for faster empty operation.
5. Disable the GMAC transmitter by clearing Bit 3 (TE) in MAC Configuration (MAC_Configuration).
6. After the link is up, read the PHY registers to know the latest configuration and accordingly program the GMAC registers.
7. Restart the operation by starting the Tx DMA, and then enabling the GMAC Transmitter and Receiver

6.5.7 Programming Guidelines for IEEE 1588 Timestamping

6.5.7.1 Initialization Guideline for System Time Generation

You can enable the timestamp feature by setting Bit 0 of the Timestamp control register. However, it is essential that the timestamp counter should be initialized after this bit is set. Complete the following steps during GMAC initialization:

1. Mask the Timestamp Trigger interrupt by setting Bit 9 of Interrupt Mask Register (Interrupt_Mask).
2. Program Bit 0 in Timestamp Control Register (Timestamp_Control) to enable timestamping.
3. Program Sub Second Increment Register (Sub_Second_Increment) based on the PTP clock frequency.
4. If you are using the Fine Correction approach, program Timestamp Addend Register (Timestamp_Addend) and set Bit 5 of Timestamp Control Register (Timestamp_Control).
5. Poll the Timestamp Control register until Bit 5 is cleared.
6. Program Bit 1 of Timestamp Control Register (Timestamp_Control) to select the Fine Update method (if required).
7. Program System Time – Seconds Update Register (System_Time_Seconds_Update) and System Time – Nanoseconds Update Register (System_Time_Nanoseconds_Update) with the appropriate time value.
8. Set Bit 2 in Timestamp Control Register (Timestamp_Control). The timestamp counter starts operation as soon as it is initialized with the value written in the Timestamp Update registers.
9. Enable the GMAC receiver and transmitter for proper timestamping.

CAUTION

If timestamp operation is disabled by clearing Bit 0 of Timestamp Control Register (Timestamp_Control), you need to repeat all these steps to restart the timestamp operation.

6.5.7.2 System Time Correction

To synchronize or update the system time in one process (coarse correction method), complete the following steps:

1. Set the offset (positive or negative) in the Timestamp Update registers (System Time – Seconds Update Register (System_Time_Seconds_Update) and System Time – Nanoseconds Update Register (System_Time_Nanoseconds_Update)).
2. Set Bit 3 (TSUPDT) of Timestamp Control Register (Timestamp_Control). The value in the Timestamp Update registers is added to or subtracted from the system time when the TSUPDT bit is cleared.

To synchronize or update the system time to reduce system time jitter (fine correction method), complete the following steps:

1. With the help of the algorithm explained in **Section 6.5.8, System Time Register Module**, calculate the rate by which you want to make the system time increments slower or faster.
2. Update Timestamp Addend Register (Timestamp_Addend) with the new value and set Bit 5 of Timestamp Control Register (Timestamp_Control).
3. Wait for the time for which you want the new value of the Addend register to be active. You can do this by enabling the Timestamp Trigger interrupt after the system time reaches the target value.
4. Program the required target time in Target Time Seconds Register (Target_Time_Seconds) and Target Time Nanoseconds Register (Target_Time_Nanoseconds).
5. Unmask the Timestamp interrupt by clearing bit 9 of Interrupt Mask Register (Interrupt_Mask).
6. Set bit 4 in Timestamp Control Register (Timestamp_Control).
7. When this trigger causes an interrupt, read Interrupt Status Register (Interrupt_Status).
8. Reprogram Timestamp Addend Register (Timestamp_Addend) with the old value and set bit 5 again.

6.5.8 System Time Register Module

The 64-bit time is maintained in this module and updated using the input reference clock (GMAC_PTP_REFCLK_I). This time is the source for taking snapshots (timestamps) of Ethernet frames being transmitted or received at the GMII.

The System Time counter can be initialized or corrected using the coarse correction method. In this method, the initial value or the offset value is written to the Timestamp Update register (see **Section 6.6.8, IEEE 1588 Timestamp Registers**). For initialization, the System Time counter is written with the value in the Timestamp 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 (GMAC_PTP_REFCLK_I) 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 coarse 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. 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.

Figure 6.3 System Time Update Using Fine 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 (GMAC_PTP_REFCLK_I) is, for example, 66 MHz, this ratio is calculated as 66 MHz / 50 MHz = 1.32. Hence, the default addend value to be set in the register is 2^{32} / 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 addend value of 0xC1F07C1F (2^{32} / 1.32) must be programmed.

In **Figure 6.3, System Time Update Using Fine Method**, the constant value used to accumulate the sub second register is decimal 43, which achieves an accuracy of 20 ns in the system time (in other words, it is incremented in 20 ns steps). The optional System Time module is unavailable when External Time Update is enabled. Two different methods are used to update the System Time register, depending on which configuration you choose.

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:

$$\mathrm{FreqCompensationValue0} = \frac{2^{32}}{\mathrm{FreqDivisionRatio}}$$

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 MasterSyncTime(n), the master sends the slave clock a Sync message. The slave receives this message when its local clock is SlaveClockTime(n) and computes MasterClockTime(n) as:

$$\mathrm{MasterClockTime(n)} = \mathrm{MasterSyncTime(n)} + \mathrm{MasterToSlaveDelay(n)}$$

- The master clock count for current Sync cycle, MasterClockCount(n) is given by:

$$\mathrm{MasterClockCount(n)} = \mathrm{MasterClockTime(n)} - \mathrm{MasterClockTime}(\mathrm{n}-1)$$
$$(\text{assuming that MasterToSlaveDelay is the same for Sync cycles n and n} - 1)$$

- The slave clock count for current Sync cycle, SlaveClockCount(n) is given by:

$$\mathrm{SlaveClockCount(n)} = \mathrm{SlaveClockTime(n)} - \mathrm{SlaveClockTime(n-1)}$$

- The difference between master and slave clock counts for current Sync cycle, ClockDiffCount(n) is given by:

$$\mathrm{ClockDiffCount(n)} = \mathrm{MasterClockCount(n)} - \mathrm{SlaveClockCount(n)}$$

- The frequency scaling factor for slave clock, FreqScaleFactor(n) is given by:

$$\mathrm{FreqScaleFactor(n)} = \frac{\mathrm{MasterClockCount(n)} + \mathrm{ClockDiffCount(n)}}{\mathrm{SlaveClockCount(n)}}$$

- The frequency compensation value for Addend register, FreqCompensationValue(n) is given by

$$\mathrm{FreqCompensationValue(n)} = \mathrm{FreqScaleFactor(n)} \times \mathrm{FreqCompensationValue}(\mathrm{n} - 1)$$

In theory, this algorithm achieves lock in one Sync cycle; however, it may take several cycles, because of 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 corrects it at the cost of more Sync cycles.

6.5.9 Programming Guidelines for Energy Efficient Ethernet

6.5.9.1 Entering and Exiting the Tx LPI Mode

Complete 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 Bits[26:16] and Bits[15:0] in LPI Timers Control Register (LPI_Timers_Control).
4. Read the link status of the PHY chip by using the MDIO interface and update Bit 17 of LPI Control and Status Register (LPI_Control_Status) accordingly. This update should be done whenever the link status in the PHY chip changes.
5. Set Bit 16 of LPI Control and Status Register (LPI_Control_Status) to make the GMAC enter the LPI state. The GMAC enters the LPI mode after completing the transmission in progress and sets Bit 0.

CAUTION

- To make the GMAC enter the LPI state only after it completes the transmission of all queued frames in the Tx FIFO, you should set Bit 19 in LPI Control and Status Register (LPI_Control_Status).
- To switch off the CSR clock (GMAC[m]_HCLK) or power to the rest of the system during the LPI state, you should wait for the TLPIEN interrupt of LPI Control and Status Register (LPI_Control_Status) to be generated. Restore the clocks before performing step 6 when you want to come out of the LPI state.

6. Reset Bit 16 of LPI Control and Status Register (LPI_Control_Status) to bring the GMAC out of the LPI state The GMAC waits for the time programmed in Bits [15:0] before setting the TLPIEX interrupt status bit and resuming the transmission.

6.5.9.2 Gating Off the CSR Clock in the LPI Mode

You can gate off the CSR clock (GMAC[m]_HCLK) to save the power when the GMAC is in the Low Power Idle (LPI) mode.

(1) Gating Off the CSR Clock in the Rx LPI Mode

The following operations are performed when the GMAC receives the LPI pattern from the PHY.

1. The MAC RX enters the LPI mode and the Rx LPI entry interrupt status [RLPIEN interrupt of LPI Control and Status Register (LPI_Control_Status)] is set.
2. The interrupt signal (GMAC[m]_SBD_Int) is asserted. The GMAC[m]_SBD_Int interrupt is cleared when the host reads the LPI Control and Status Register (LPI_Control_Status).

After the GMAC[m]_SBD_Int interrupt is asserted and the MAC TX is also in the LPI mode, you can gate off the CSR clock. If the MAC TX is not in the LPI mode when you gate off the CSR clock, the events on the GMAC transmitter do not get reported or updated in the CSR.

For restoring the CSR clock, wait for the LPI exit indication from the PHY after which the GMAC asserts the LPI exit interrupt on GMAC[m]_LPI_Int (synchronous to Receive Clock). The GMAC[m]_LPI_Int interrupt is cleared when LPI_Control_Status is read.

(2) Gating Off the CSR Clock in the Tx LPI Mode

The following operations are performed when Bit 16 (LPIEN) of LPI Control and Status Register (LPI_Control_Status) is set:

1. The Transmit LPI Entry interrupt (TLPIEN bit of register LPI_Control_Status) is set.
2. The interrupt signal (GMAC[m]_SBD_Int) is asserted. The GMAC[m]_SBD_Int interrupt is cleared when the host reads the register LPI_Control_Status.

After the GMAC[m]_SBD_Int interrupt is asserted and the MAC RX is also in the LPI mode, you can gate off the CSR clock. If the MAC RX is not in the LPI mode when you gate off the CSR clock, the events on the GMAC receiver do not get reported or updated in the CSR.

For restoring the CSR clock, switch on the CSR clock when the GMAC has to come out of the TX LPI mode. After the CSR clock is resumed, reset Bit 16 (LPIEN) of LPI Control and Status Register (LPI_Control_Status) to bring the GMAC out of the LPI mode.

6.5.10 Programming Guidelines for Flexible Pulse Per Second (PPS) Output (GMAC1 only)

6.5.10.1 Generating Single Pulse on PPS

To generate single Pulse on PPS:

1. Program 2’b11 or 2’b10 (for interrupt) in Bits [6:5], TRGTMODSEL, of PPS Control Register (PPS_Control). This instructs the GMAC to use the Target Time registers (Target_Time_Seconds and Target_Time_Nanoseconds) for start time of PPS signal output.
2. Program the start time value in the Target Time registers (Target_Time_Seconds and Target_Time_Nanoseconds).
3. Program the width of the PPS signal output in PPS0 Width Register (PPS0_Width).
4. Program Bits [3:0], PPSCMD, of PPS Control Register (PPS_Control) to 4’b0001. This instructs the GMAC to generate single pulse on the PPS signal output at the time programmed in the Target Time registers (Target_Time_Seconds and Target_Time_Nanoseconds).

When the PPSCMD is executed (PPSCMD bits = 0), you can cancel the pulse generation by giving the Cancel Start Command (PPSCMD = 4’b0011) before the programmed start time elapses. You can also program the behavior of the next pulse in advance. To program the next pulse:

1. Program the start time for the next pulse in the Target Time registers (Target_Time_Seconds and Target_Time_Nanoseconds). This time should be more than the time at which the falling edge occurs for the previous pulse.
2. Program the width of the next PPS signal output in PPS0 Width Register (PPS0_Width).
3. Program Bits [3:0], PPSCMD, of PPS Control Register (PPS_Control) to generate a single pulse after the time at which the previous pulse is de-asserted. This instructs the GMAC to generate single pulse on the PPS signal output, at the time programmed in Target Time registers.

If you give this command before the previous pulse becomes low, then the new command overwrites the previous command and the GMAC may generate only 1 extended pulse.

6.5.10.2 Generating a Pulse Train on PPS

To generate a pulse train on PPS:

1. Program 2’b11 or 2’b10 (for interrupt) in Bits [6:5], TRGTMODSEL, of PPS Control Register (PPS_Control). This instructs the GMAC to use the Target Time registers (Target_Time_Seconds and Target_Time_Nanoseconds) for start time of the PPS signal output.
2. Program the start time value in the Target Time registers (Target_Time_Seconds and Target_Time_Nanoseconds).
3. Program the interval value between the train of pulses on the PPS signal output in PPS0 Width Register (PPS0_Width).
4. Program the width of the PPS signal output in PPS0 Width Register (PPS0_Width).
5. Program Bits[3:0], PPSCMD, of PPS Control Register (PPS_Control) to 4’b0010. This instructs the GMAC to generate train of pulses on the PPS signal output with start time programmed in Target Time registers (Target_Time_Seconds and Target_Time_Nanoseconds).

By default, the PPS pulse train is free running unless stopped by “STOP Pulse train at time” or “STOP Pulse Train immediately” commands.

6. Program the stop value in the Target Time registers (Target_Time_Seconds and Target_Time_Nanoseconds). Ensure that Bit 31 (TRGTBUSY) of Target Time Nanoseconds Register (Target_Time_Nanoseconds) is reset before programming the Target Time registers (Target_Time_Seconds and Target_Time_Nanoseconds) again.
7. Program the PPSCMD field (bit 3:0) of PPS Control Register (PPS_Control) to 4’b0100. This stops the train of pulses on PPS signal output after the programmed stop time specified in Step 6 elapses.

You can stop the pulse train at any time by programming 4’b0101 in the PPSCMD field. Similarly, you can cancel the Stop Pulse train command (given in Step 7) by programming 4’b0110 in the PPSCMD field before the time (programmed in Step 6) elapses. You can cancel the pulse train generation by programming 4’b0011 in the PPSCMD field before the programmed start time (in Step 2) elapses.

6.5.10.3 Generating an Interrupt without Affecting the PPS

The Bits [6:5], TRGTMODSEL, of PPS Control Register (PPS_Control) enable you to program the Target Time registers (Target_Time_Seconds and Target_Time_Nanoseconds) to do any one of the following:

(A) Generate only interrupts.

(B) Generate interrupts and the PPS start and stop time.

(C) Generate only PPS start and stop time.

To program the Target Time registers (Target_Time_Seconds and Target_Time_Nanoseconds) to generate only interrupt event:

1. Program 2’b00 (for interrupt) in Bits [6:5], TRGTMODSEL, of PPS Control Register (PPS_Control). This instructs the GMAC to use the Target Time registers (Target_Time_Seconds and Target_Time_Nanoseconds) for target time interrupt.
2. Program a target time value in the Target Time registers (Target_Time_Seconds and Target_Time_Nanoseconds). This instructs the GMAC to generate an interrupt when the target time elapses.

If Bits [6:5], TRGTMODSEL, are changed (for example, to control the PPS), then the interrupt generation is over written with the new mode and new programmed Target Time register value.

6.5.11 Enhanced Descriptors

The integrated DMA transfers data based on a linked list of descriptors. The descriptors are created in the system memory. Descriptor contains two buffers, two byte-count buffers, and two address pointers, which enable the adapter port to be compatible with various types of memory management schemes.

CAUTION

There is no limit for number of descriptors that can be used for a single frame.

The enhanced descriptor structure can have 8 DWORDS (32-byte). The features of the enhanced descriptor structure are:

- The enhanced descriptor structure is implemented to support buffers of up to 8 KB (useful for Jumbo frames).
- There is an assignment of control and status bits in TDES0, TDES1, RDES0, and RDES1.
- The transmit descriptor stores the timestamp in TDES6 and TDES7.
- This receive descriptor structure is also used for storing the extended status (RDES4) and timestamp (RDES6 and RDES7).
- You can select one of the following options for descriptor structure:
  - If timestamping is enabled in Timestamp Control Register (Timestamp_Control) or Checksum Offload is enabled in MAC Configuration Register (MAC_Configuration), the software needs to allocate 32-byte (8 DWORDS) of memory for every descriptor. For this, the software should set Bit 7 (Enhanced Descriptor Size) of Bus Mode Register (Bus_Mode).
  - If timestamping and Checksum Offload is not enabled, the extended descriptors (DES4 to DES7) are not required. Therefore, the software can use enhanced descriptors with the default size of 16 bytes (4 DWORDS). For this, the software should reset Bit 7 (Enhanced Descriptor Size) of Bus Mode Register (Bus_Mode) to 0.

6.5.11.1 Transmit Descriptor

The transmit descriptor structure is shown in the following table. The application software must program the control bits TDES0[31:18] during descriptor initialization. When the DMA updates the descriptor, it write backs all the control bits except the OWN bit (which it clears) and updates the status bits[7:0]. The contents of the transmitter descriptor word 0 (TDES0) through word 3 (TDES3) are given in **Table 6.120** through **Table 6.123**, respectively.

With the advance timestamp support, the snapshot of the timestamp to be taken can be enabled for a given frame by setting Bit 25 (TTSE) of TDES0. When the descriptor is closed (that is, when the OWN bit is cleared), the timestamp is written into TDES6 and TDES7. This is indicated by the status Bit 17 (TTSS) of TDES0 shown in the following table. The contents of TDES6 and TDES7 are mentioned in **Table 6.124** and **Table 6.125**.

CAUTION

When Advanced Timestamp feature is enabled, the software should set Bit 7 of Bus Mode Register (Bus_Mode), so that the DMA operates with extended descriptor size. When this control bit is reset, the TDES4 to TDES7 descriptor space is not valid.

Table 6.118 Transmit Descriptor Fields - Enhanced Format

	31	30 29 28 27 26	25	24 23 22 21 20 19 18	17	16 15 14 13 12 11 10 9 8 7	6 5 4 3	2 1 0
TDES0	OWN	Ctrl [30:26]	TTSE	Ctrl [24:18]	TTSS	Status [16:7]	Ctrl/Status [6:3]	Status [2:0]
TDES1	Reserved	Buffer 2 Byte Count [28:16]				Reserved	Buffer 1 Byte Count [12:0]	
TDES2	Buffer 1 Address [31:0]							
TDES3	Buffer 2 Address [31:0] or Next Descriptor Address [31:0]							
TDES4	Reserved							
TDES5	Reserved							
TDES6	Transmit Timestamp Low [31:0]							
TDES7	Transmit Timestamp High [31:0]							

The DMA always reads or fetches four DWORDS of the descriptor from system memory to obtain the buffer and control information as shown in the following table.

Table 6.119 Transmit Descriptor Fetch (Read) for Enhanced Format

	31	30 29 28 27 26	25	24 23 22 21 20 19 18	17 16 15 14 13 12 11 10 9 8 7	6 5 4 3	2 1 0
TDES0	OWN	Ctrl [30:26]	TTSE	Ctrl [24:18]	Reserved for Status [17:7]	Reserved	Reserved for Status [2:0]
TDES1	Reserved	Buffer 2 Byte Count [28:16]			Reserved	Buffer 1 Byte Count [12:0]	
TDES2	Buffer 1 Address [31:0]						
TDES3	Buffer 2 Address [31:0] or Next Descriptor Address [31:0]						

Table 6.120 Transmit Descriptor Word 0 (TDES0) (1/3)

Bit Position	Bit Name	Description
b31	OWN	Own Bit When set, this bit indicates that the descriptor is owned by the DMA. When this bit is reset, it 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 read completely. The ownership bit of the frame's first descriptor must 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 an ownership bit.
b30	IC	Interrupt on Completion When set, this bit sets the Transmit Interrupt (Bit 0 of Status Register) after the present frame has been transmitted. This bit is valid only when the last segment bit (TDES0[29]) is set.
b29	LS	Last Segment When set, this bit indicates that the buffer contains the last segment of the frame. When this bit is set, the TBS1 or TBS2 field in TDES1 should have a non-zero value.
b28	FS	First Segment When set, this bit indicates that the buffer contains the first segment of a frame.
b27	DC	Disable CRC When this bit is set, the GMAC does not append a cyclic redundancy check (CRC) to the end of the transmitted frame. This is valid only when the first segment (TDES0[28]) is set.
b26	DP	Disable Pad When set, the GMAC does not automatically add padding to a frame shorter than 64 bytes. When this bit is reset, the DMA automatically adds padding and CRC to a frame shorter than 64 bytes, and the CRC field is added despite the state of the DC bit (TDES0[27]). This is valid only when the first segment (TDES0[28]) is set.
b25	TTSE	Transmit Timestamp Enable When set, this bit enables IEEE 1588 hardware timestamping for the transmit frame referenced by the descriptor. This field is valid only when the First Segment control bit (TDES0[28]) is set.
b24	Reserved	Reserved
b23 to b22	CIC	Checksum Insertion Control These bits control the checksum calculation and insertion. The following list describes the bit encoding: 2'b00: Checksum Insertion Disabled. 2'b01: Only IP header checksum calculation and insertion are enabled. 2'b10: IP header checksum and payload checksum calculation and insertion are enabled, but pseudo header checksum is not calculated in hardware. 2'b11: IP Header checksum and payload checksum calculation and insertion are enabled, and pseudo header checksum is calculated in hardware. This field is valid when the First Segment control bit (TDES0[28]) is set.
b21	TER	Transmit End of Ring When 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.
b20	TCH	Second Address Chained When set, this bit indicates that the second address in the descriptor is the Next Descriptor address rather than the second buffer address. When TDES0[20] is set, TBS2 (TDES1[28:16]) is a "don't care" value. TDES0[21] takes precedence over TDES0[20].
b19 to b18	Reserved	Reserved
b17	TTSS	Transmit Timestamp Status This field is used as a status bit to indicate that a timestamp was captured for the described transmit frame. When this bit is set, TDES2 and TDES3 have a timestamp value captured for the transmit frame. This field is only valid when the descriptor's Last Segment control bit (TDES0[29]) is set.

Table 6.120 Transmit Descriptor Word 0 (TDES0) (2/3)

Bit Position	Bit Name	Description
b16	IHE	IP Header Error When 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. If COE detects an IP header error, it still inserts an IPv4 header checksum if the Ethernet Type field indicates an IPv4 payload.
b15	ES	Error Summary Indicates the logical OR of the following bits: TDES0[14]: Jabber Timeout TDES0[13]: Frame Flush 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 TDES0[16]: IP Header Error TDES0[12]: IP Payload Error
b14	JT	Jabber Timeout When set, this bit indicates the GMAC transmitter has experienced a jabber timeout. This bit is only set when Bit 22 (Jabber Disable) of MAC Configuration Register (MAC_Configuration) is not set.
b13	FF	Frame Flushed When set, this bit indicates that the DMA or MTL flushed the frame because of a software Flush command given by the CPU.
b12	IPE	IP Payload Error When 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.
b11	LOC	Loss of Carrier When set, this bit indicates that a 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 valid only for the frames transmitted without collision when the GMAC operates in the half-duplex mode.
b10	NC	No Carrier When set, this bit indicates that the Carrier Sense signal from the PHY was not asserted during transmission.
b9	LC	Late Collision When set, this bit indicates that frame transmission is aborted because of a collision occurring after the collision window (64 byte-times, including preamble, in MII mode and 512 byte-times, including preamble and carrier extension, in GMII mode). This bit is not valid if the Underflow Error bit is set.
b8	EC	Excessive Collision When set, this bit indicates that the transmission was aborted after 16 successive collisions while attempting to transmit the current frame. If Bit 9 (Disable Retry bit) in the MAC Configuration Register (MAC_Configuration) is set, this bit is set after the first collision, and the transmission of the frame is aborted.
b7	VF	VLAN Frame When set, this bit indicates that the transmitted frame is a VLAN-type frame.

Table 6.120 Transmit Descriptor Word 0 (TDES0) (3/3)

Bit Position	Bit Name	Description
b6 to b3	CC	Collision Count (Status field) These status bits indicate the number of collisions that occurred before the frame was transmitted. This count is not valid when the Excessive Collisions bit (TDES0[8]) is set. The core updates this status field only in the half-duplex mode.
b2	ED	Excessive Deferral When set, this bit indicates that the transmission has ended because of excessive deferral of over 24,288-bit times (155,680-bit times in 1000 Mbps mode or if Jumbo Frame is enabled) if Bit 4 (Deferral Check bit) in MAC Configuration Register (MAC_Configuration) is set high.
b1	UF	Underflow Error When set, this bit indicates that the GMAC aborted the frame because the 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 Bit 5 (Transmit Underflow bit) in Status Register (Status) and Bit 0 (Transmit Interrupt bit) in Status Register (Status).
b0	DB	Deferred Bit When set, this bit indicates that the GMAC defers before transmission because of the presence of carrier. This bit is valid only in the half-duplex mode.

Table 6.121 Transmit Descriptor Word 1 (TDES1)

Bit Position	Bit Name	Description
b31 to b29	Reserved	Reserved
b28 to b16	TBS2	Transmit Buffer 2 Size This field indicates the second data buffer size in bytes. This field is not valid if TDES0[20] is set. 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 TDES1 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 TDES1), then the DMA marks the last transfer from that data buffer as the end-of frame to the MTL.
b15 to b13	Reserved	Reserved
b12 to b0	TBS1	Transmit Buffer 1 Size These bits indicate the first data buffer byte size, in bytes. If this field is 0, the DMA ignores this buffer and uses Buffer 2 or the next descriptor, depending on the value of TCH (TDES0[20]).

Table 6.122 Transmit Descriptor Word 2 (TDES2)

Bit Position	Bit Name	Description
b31 to b0	Buffer 1 Address Pointer	Buffer 1 Address Pointer These bits indicate the physical address of Buffer 1. There is no limitation on the buffer address alignment.

Table 6.123 Transmit Descriptor Word 3 (TDES3)

Bit Position	Bit Name	Description
b31 to b0	Buffer 2 Address Pointer or Next Descriptor Address	Buffer 2 Address Pointer (Next Descriptor Address) Indicates 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 contains the pointer to 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.)

Table 6.124 Transmit Descriptor Word 6 (TDES6)

Bit Position	Bit Name	Description
b31 to b0	TTSL	Transmit Frame Timestamp Low This field is updated by DMA with the least significant 32 bits of the timestamp captured for the corresponding transmit frame. This field has the timestamp only if the Last Segment bit (LS) in the descriptor is set and Timestamp status (TTSS) bit is set.

Table 6.125 Transmit Descriptor Word 7 (TDES7)

Bit Position	Bit Name	Description
b31 to b0	TTSH	Transmit Frame Timestamp High This field is updated by DMA with the most significant 32 bits of the timestamp captured for the corresponding transmit frame. This field has the timestamp only if the Last Segment bit (LS) in the descriptor is set and Timestamp status (TTSS) bit is set.

6.5.11.2 Receive Descriptor

The structure of the received descriptor is shown in the following table. This can have 32 bytes of descriptor data (8 DWORDs).

CAUTION

When either of these features is enabled, the Software should set Bit 7 of Bus Mode Register (Bus_Mode) so that the DMA operates with extended descriptor size. When this control bit is reset, the RDES0[0] is always cleared and the RDES4 to RDES7 descriptor space is not valid.

Table 6.126 Receive Descriptor Fields - Enhanced Format

	31	30 29	28 27 26 25 24 23 22 21 20 19 18 17 16	15 14	13	12 11 10 9 8 7 6 5 4 3 2 1 0
RDES0	OWN	Status [30:0]				
RDES1	CTRL	RES [30:29]	Buffer 2 Byte Count [28:16]	CTRL [15:14]	RES	Buffer 1 Byte Count [12:0]
RDES2	Buffer 1 Address [31:0]					
RDES3	Buffer 2 Address [31:0] or Next Descriptor Address [31:0]					
RDES4	Extended Status [31:0]					
RDES5	Reserved					
RDES6	Receive Timestamp Low [31:0]					
RDES7	Receive Timestamp High [31:0]					

The contents of RDES0 are identified in **Table 6.127**. The contents of RDES1 through RDES3 are identified in **Table 6.128** through**, Table 6.130** respectively.

CAUTION

The status bits are valid when the LS bit is set. Therefore, when multiple descriptors are used for a single frame, status bits of only last descriptor (that is, descriptor with LS bit set) are valid.

Table 6.127 Receive Descriptor Fields 0 (RDES0) (1/3)

Bit Position	Bit Name	Description
b31	OWN	Own Bit When set, this bit indicates that the descriptor is owned by the DMA of the GMAC. 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.
b30	AFM	Destination Address Filter Fail When set, this bit indicates a frame that failed in the DA Filter in the GMAC.
b29 to b16	FL	Frame Length These bits indicate the byte length of the received frame that was transferred to host memory. This field is valid when Last Descriptor (RDES0[8]) is set and either the Descriptor Error (RDES0[14]) or Overflow Error (RDES0[11]) bits are reset. The frame length also includes the two bytes appended to the Ethernet frame when IP checksum calculation (Type 1) is enabled and the received frame is not a MAC control frame. This field is valid 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. The inclusion of CRC length in the frame length depends on the settings of Bit 7 and Bit 25 in MAC Configuration Register (MAC_Configuration).

Table 6.127 Receive Descriptor Fields 0 (RDES0) (2/3)

Bit Position	Bit Name	Description
b15	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]: Giant Frame RDES4[4:3]: IP Header or Payload Error RDES0[11]: Overflow Error RDES0[14]: Descriptor Error This field is valid only when the Last Descriptor (RDES0[8]) is set.
b14	DE	Descriptor Error When 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 Last Descriptor (RDES0[8]) is set.
b13	SAF	Source Address Filter Fail When set, this bit indicates that the SA field of frame failed the SA Filter in the GMAC.
b12	LE	Length Error When set, this bit indicates that the actual length of the frame received and that the Length/Type field does not match. This bit is valid only when the Frame Type (RDES0[5]) bit is reset.
b11	OE	Overflow Error When set, this bit indicates that the received frame was damaged because of buffer overflow in MTL. ***Note)*** This bit is set only when the DMA transfers a partial frame to the application. This happens only when the Rx FIFO is operating in the threshold mode. In the store-and-forward mode, all partial frames are dropped completely in Rx FIFO.
b10	VLAN	VLAN Tag When set, this bit indicates that the frame to which this descriptor is pointing is a VLAN frame tagged by the GMAC. The VLAN tagging depends on checking the VLAN fields of received frame based on the VLAN Tag Register (VLAN_Tag) setting.
b9	FS	First Descriptor When 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.
b8	LS	Last Descriptor When set, this bit indicates that the buffers pointed to by this descriptor are the last buffers of the frame.
b7	Timestamp Available or Giant Frame	Timestamp Available or Giant Frame When 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. Otherwise, this bit, when set, indicates the Giant Frame Status. Giant frames are larger than 1,518-byte (or 1,522-byte for VLAN or 2,000-byte when Bit 27 of MAC Configuration register is set) normal frames and larger than 9,018-byte (9,022-byte for VLAN) frame when Jumbo Frame processing is enabled.
b6	LC	Late Collision When set, this bit indicates that a late collision has occurred while receiving the frame in the half-duplex mode.
b5	FT	Frame Type When set, this bit indicates that the Receive Frame is an Ethernet-type frame (the Length/Type field is greater than or equal to 1,536). When this bit is reset, it indicates that the received frame is an IEEE 802.3 frame. This bit is not valid for Runt frames less than 14 bytes.

Table 6.127 Receive Descriptor Fields 0 (RDES0) (3/3)

Bit Position	Bit Name	Description
b4	RWT	Receive Watchdog Timeout When 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.
b3	RE	Receive Error When set, this bit indicates that the GMII/MII RXER signal is asserted while GMII/MII RXDV is asserted during frame reception. This error also includes carrier extension error in the GMII and half-duplex mode. Error can be of less or no extension, or error (rxd != 0xf) during extension.
b2	DE	Dribble Bit Error When set, this bit indicates that the received frame has a non-integer multiple of bytes (odd nibbles). This bit is valid only in the MII Mode.
b1	CE	CRC Error When 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.
b0	Extended Status Available	Extended Status Available When set, this bit indicates that the extended status is available in descriptor word 4 (RDES4). This is valid only when the Last Descriptor bit (RDES0[8]) is set. This bit is invalid when Bit 30 is set. This bit is set even when IP Checksum Offload engine bypasses the processing of received frame. The bypassing may be because of non-IP frame or IP frame with non-TCP/UDP/ICMP payload.

Table 6.128 Receive Descriptor Fields 1 (RDES1)

Bit Position	Bit Name	Description
b31	DIC	Disable Interrupt on Completion When set, this bit prevents setting the Status Register's RI bit (Bit 6) for the received frame ending in the buffer indicated by this descriptor. This, in turn, disables the assertion of the interrupt to Host because of RI for that frame. ***Note*)** This bit is valid only when the last descriptor bit (RDES0[8]) is set.
b30 to b29	Reserved	Reserved
b28 to b16	RBS2	Receive Buffer 2 Size These bits indicate the second data buffer size, 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. If the buffer size is not an appropriate multiple of 4, the resulting behavior is undefined. This field is not valid if RDES1[14] is set.
b15	RER	Receive End of Ring When 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.
b14	RCH	Second Address Chained When set, this bit indicates that the second address in the descriptor is the Next Descriptor address rather than the second buffer address. When this bit is set, RBS2 (RDES1[28:16]) is a "don't care" value. RDES1[15] takes precedence over RDES1[14].
b13	Reserved	Reserved
b12 to b0	RBS1	Receive Buffer 1 Size Indicates the first data buffer size in bytes. The buffer size must be a multiple of 4, even if the value of RDES2 (buffer1 address pointer) is not aligned. 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 (Bit 14).
b31 to b0	Buffer 1 Address Pointer	Buffer 1 Address Pointer These bits indicate the physical address of Buffer 1. There are no limitations on the buffer address alignment except for the following condition: The DMA uses the configured value for its address generation when the RDES2 value is used to store the start of frame. The DMA performs a write operation with the RDES2[1:0] bits as 0 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.

Table 6.129 Receive Descriptor Fields 2 (RDES2)

Bit Position	Bit Name	Description
b31 to b0	Buffer 1 Address Pointer	Buffer 1 Address Pointer These bits indicate the physical address of Buffer 1. There are no limitations on the buffer address alignment except for the following condition: The DMA uses the configured value for its address generation when the RDES2 value is used to store the start of frame. The DMA performs a write operation with the RDES2[1:0] bits as 0 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.

Table 6.130 Receive Descriptor Fields 3 (RDES3)

Bit Position	Bit Name	Description
b31 to b0	Buffer 2 Address Pointer or Next Descriptor Address	Buffer 2 Address Pointer (Next Descriptor Address) These bits indicate 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 contains the pointer to the physical memory where the Next Descriptor is present. If RDES1[14] is set, the buffer (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 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 store the start of frame. The DMA ignores RDES3[1:0] if the address pointer is to a buffer where the middle or last part of the frame is stored.

The extended status written is as shown in the following table. The extended status is written only when there is status related to IPC or timestamp available. The availability of extended status is indicated by Bit 0 of RDES0.

Table 6.131 Receive Descriptor Fields 4 (RDES4) (1/2)

Bit Position	Bit Name	Description
b31 to b15	Reserved	Reserved
b14	Timestamp Dropped	Timestamp Dropped When set, this bit indicates that the timestamp was captured for this frame but got dropped in the MTL Rx FIFO because of overflow.
b13	PTP Version	PTP Version When set, this bit indicates that the received PTP message is having the IEEE 1588 version 2 format. When reset, it has the version 1 format.
b12	PTP Frame Type	PTP Frame Type When set, this bit indicates that the PTP message is sent directly over Ethernet. When this bit is not set and the message type is non-zero, it indicates that the PTP message is sent over UDP-IPv4 or UDP-IPv6. The information about IPv4 or IPv6 can be obtained from Bits 6 and 7.
b11 to b8	Message Type	Message Type These bits are encoded to give the type of the message received. 4’b0000: No PTP message received 4’b0001: SYNC (all clock types) 4’b0010: Follow_Up (all clock types) 4’b0011: Delay_Req (all clock types) 4’b0100: Delay_Resp (all clock types) 4’b0101: Pdelay_Req (in peer-to-peer transparent clock) 4’b0110: Pdelay_Resp (in peer-to-peer transparent clock) 4’b0111: Pdelay_Resp_Follow_Up (in peer-to-peer transparent clock) 4’b1000: Announce 4’b1001: Management 4’b1010: Signaling 4’b1011 to 4’b1110: Reserved 4’b1111: PTP packet with Reserved message type

Table 6.131 Receive Descriptor Fields 4 (RDES4) (2/2)

Bit Position	Bit Name	Description
b7	IPv6 Packet Received	IPv6 Packet Received When set, this bit indicates that the received packet is an IPv6 packet. This bit is updated only when Bit 10 (IPC) of MAC Configuration Register (MAC_Configuration) is set.
b6	IPv4 Packet Received	IPv4 Packet Received When set, this bit indicates that the received packet is an IPv4 packet. This bit is updated only when Bit 10 (IPC) of MAC Configuration Register (MAC_Configuration) is set.
b5	IP Checksum Bypassed	IP Checksum Bypassed When set, this bit indicates that the checksum offload engine is bypassed.
b4	IP Payload Error	IP Payload Error When set, this bit indicates that the 16-bit IP payload checksum (that is, the TCP, UDP, or ICMP checksum) that the core 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. This bit is valid when either Bit 7 or Bit 6 is set.
b3	IP Header Error	IP Header Error When set, this bit indicates that either the 16-bit IPv4 header checksum calculated by the core does not match the received checksum bytes, or the IP datagram version is not consistent with the Ethernet Type value. This bit is valid when either Bit 7 or Bit 6 is set.
b2 to b0	IP Payload Type	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 3'b000 if it does not process the IP datagram's payload due to an IP header error or fragmented IP. 3'b000: Unknown or did not process IP payload 3'b001: UDP 3'b010: TCP 3'b011: ICMP 3'b1xx: Reserved This bit is valid when either Bit 7 or Bit 6 is set.

RDES6 and RDES7 contain the snapshot of the timestamp. The availability of the snapshot of the timestamp in RDES6 and RDES7 is indicated by Bit 7 in the RDES0 descriptor. The contents of RDES6 and RDES7 are identified in **Table 6.132** and **Table 6.133**, respectively.

Table 6.132 Receive Descriptor Fields 6 (RDES6)

Bit Position	Bit Name	Description
b31 to b0	RTSL	Receive Frame Timestamp Low This field is updated by DMA with the least significant 32 bits of the timestamp captured for the corresponding receive frame. This field is updated by DMA only for the last descriptor of the receive frame which is indicated by Last Descriptor status bit (RDES0[8]).

Table 6.133 Receive Descriptor Fields 7 (RDES7)

Bit Position	Bit Name	Description
b31 to b0	RTSH	Receive Frame Timestamp High This field is updated by DMA with the most significant 32 bits of the timestamp captured for the corresponding receive frame. This field is updated by DMA only for the last descriptor of the receive frame which is indicated by Last Descriptor status bit (RDES0[8]).

6.6 Usage Notes

6.6.1 Configurations

The configuration depends on implementation is shown here.

Clock

- CSR clock: AHB clock (GMAC[m]_HCLK)

Features

- System Interface Configuration: GMAC-AXI
- IEEE 1588 Timestamp feature: Enabled
- IEEE 1588 Auxiliary Snapshot feature: Enabled
- Advanced Timestamp feature: Enabled
- Enable Flexible Pulse Per Second Output feature: Enabled on GMAC1 only
- IP Checksum Offload feature: Enabled
- Full Checksum Offload (Type 2) features: Enabled
- Power Management feature: Enabled
- AV feature: Disabled
- Hash Table width: 256

HW_Feature Register also shows current configuration.

6.6.2 Restriction

6.6.2.1 Ethernet MAC

When any Register content is being transferred to a different clock domain after a write operation, there should not be any further writes to the same location until the first write is updated. Otherwise, the second write operation does not get updated to the destination clock domain. Therefore, the delay between two writes to the same register location should be at least 4 cycles of the destination clock (PHY Receive clock, PHY Transmit clock, or PTP clock).

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 write transfers, with address aligned to the bus width, with dummy data (old data) in the byte lanes that are not valid. This typically happens during the transfer of the beginning or end of an Ethernet frame. The software driver should discard the dummy bytes based on the start address of the buffer and size of the frame.

6.6.3 MAC Management Counters

The counters in the MAC Management Counters (MMC) module can be viewed as an extension of the register address space of the CSR (Control & Status Registers) module. The MMC module maintains a set of registers for gathering statistics on the received and transmitted frames. The register set includes a control register for controlling the behavior of the registers, two 32-bit registers containing interrupts generated (receive and transmit), and two 32-bit registers containing masks for the Interrupt register (receive and transmit). These registers are accessible from the Application through the MAC Control Interface (MCI). Each register is 32 bits width. The write data is qualified with the corresponding byte enable signals. Therefore, non 32-bit accesses are allowed as long as the address is word aligned. The MMCs are accessed using transactions, in the same way the CSR address space is accessed.

The MMC counters are free running. There is no separate enable for the counters to start. It starts counting when corresponding frame is received or transmitted. The Receive MMC counters are updated for frames that are passed by the Address Filter (AFM) block. The statistics of frames, dropped by the AFM module, are not updated unless they are runt frames of less than 6 bytes (DA bytes are not received fully). To get statistics of all frames, you should set Bit 0 in register MAC_Frame_Filter (MAC Frame Filter register).

6.6.3.1 Address Assignments

The following definitions define the terminology.

- Transmitted frames are considered “good” if transmitted successfully. In other words, a transmitted frame is good if the frame transmission is not aborted because of any of the following errors:
  - Jabber Timeout
  - No Carrier or 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 1518 bytes)
  - GMII_RXER Input error
- The maximum frame size depends on the frame type, as follows:
  - Untagged frame maxsize = 1518
  - VLAN Frame maxsize = 1522
  - Envelope Frames maxsize = 2000 (when programmed in Bit 27 of MAC Configuration Register (MAC_Configuration))

CAUTION

The MMC counters registers at the following offset addresses are of type Read-Only and have the default value of 0:

- 0x0114 to 0x0178
- 0x0180 to 0x01E4
- 0x0210 to 0x0244
- 0x0250 to 0x0284

The RX MMC counters and RX IPC MMC counters are updated only for frames that pass the Destination Address filter except when Bit 8 (UCDBC) in MMC Control Register (MMC_Control) is set for broadcast frames.

6.6.4 MMC Registers

MMC Registers are in 4400 0100h to 4400 01E4h (GMAC1) and 4400 2100h to 4400 21E4h (GMAC2). See **Section 6.3, Register Map**.

6.6.4.1 MMC Control Register (MMC_Control)

The MMC Control register establishes the operating mode of the management counters.

CAUTION

Bit 0 (Counters Reset) has higher priority than Bit 4 (Counter Preset). Therefore, when the Software tries to set both bits in the same write cycle, all counters are cleared and Bit 4 is not set.

6.6.4.2 MMC Receive Interrupt Register (MMC_Receive_Interrupt)

The MMC Receive Interrupt register maintains the interrupts that are generated when the following happens:

- Receive statistic counters reach half of their maximum values (0x8000_0000 for 32-bit counter and 0x8000 for 16-bit counter).
- Receive statistic counters cross their maximum values (0xFFFF_FFFF for 32-bit counter and 0xFFFF for 16-bit counter).

When the 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 width 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.

6.6.4.3 MMC Transmit Interrupt Register (MMC_Transmit_Interrupt)

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 cross 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 width 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.

6.6.4.4 MMC Receive Interrupt Mask Register (MMC_Receive_Interrupt_Mask)

The MMC Receive Interrupt Mask register maintains the masks for the interrupts generated when receive statistic counters reach half of their maximum value or the maximum values. This register is 32 bits width.

6.6.4.5 MMC Transmit Interrupt Mask Register (MMC_Transmit_Interrupt_Mask)

The MMC Transmit Interrupt Mask register maintains the masks for the interrupts generated when the transmit statistic counters reach half of their maximum value or the maximum values. This register is 32 bits width.

6.6.5 Power Management Block

Power Management (PMT) block Registers are in 4400 0028h to 4400 002Ch (GMAC1) and 4400 2028h to 4400 202Ch (GMAC2). See **Section 6.3, Register Map**.

The power management (PMT) block supports the reception of network (remote) wake-up frames and magic packet frames. The PMT block does not perform the clock gate function, but generates interrupts for remote wake-up frames and magic packets that the GMAC receives.

CAUTION

- The magic packet feature is implemented based on the Magic Packet Technology white paper.
- The remote wake-up frame feature is implemented based on the Device Class Power Management Reference Specification and various implementation-specific white papers.

When you enable the power down mode in the PMT block, the GMAC drops all received frames and does not forward any frame to the MTL RxFIFO or the application. The GMAC comes out of the power down mode only when a magic packet or a remote wake-up frame is received and the corresponding detection is enabled.

The PMT block is available in the receive path of GMAC. You can select both types of power management frames (remote wake-up frame and magic packet). You can use Bit 2 and Bit 1 of PMT Control and Status Register (PMT_Control_Status) to generate power management events. The application should program these bits. You can access the PMT registers in the similar manner as you access the GMAC CSR registers. For mapping information, see Remote Wake-Up Frame Filter Register (Remote_Wake_Up_Frame_Filter) and PMT Control and Status Register (PMT_Control_Status).

6.6.5.1 PMT Block Description

(1) Remote Wake-Up Frame Filter Register

The register Remote_Wake_Up_Frame_Filter (address: 0028h) accesses the Wake-up Frame Filter register (wkupfmfilter_reg[n], n = 0..7). To write to Wake-up Frame Filter register, write to 0028h in order to wkupfmfilter_reg0, wkupfmfilter_reg1, ... wkupfmfilter_reg7 at eight times. Read is in a similar way. The GMAC updates the current pointer value of wkupfmfilter_reg[n] register in Bits[26:24] of PMT Control and Status Register (PMT_Control_Status).

CAUTION

- If you are accessing these registers in byte or half-word mode, the internal counter, to access the appropriate wkupfmfilter_reg, is incremented when CPU accesses the lane3.
- When any Register content is being transferred to a different clock domain after a write operation, there should not be any further writes to the same location until the first write is updated. Otherwise, the second write operation does not get updated to the destination clock domain. Therefore, the delay between two writes to the same register location should be at least 4 cycles of the destination clock (PHY receive clock, PHY transmit clock, or PTP clock).

Table 6.134 Wake-Up Frame Filter Register

wkupfmfilter_reg0	Filter0 Byte Mask							
wkupfmfilter_reg1	Filter1 Byte Mask							
wkupfmfilter_reg2	Filter2 Byte Mask							
wkupfmfilter_reg3	Filter3 Byte Mask							
wkupfmfilter_reg4	RSVD	Filter3 Command	RSVD	Filter2 Command	RSVD	Filter1 Command	RSVD	Filter0 Command
wkupfmfilter_reg5	Filter3 Offset		Filter2 Offset		Filter1 Offset		Filter0 Offset	
wkupfmfilter_reg6	Filter1 CRC-16				Filter0 CRC-16			
wkupfmfilter_reg7	Filter3 CRC-16				Filter2 CRC-16			

Filter i Byte Mask

The Filter i Byte Mask register defines the bytes of the frame that are examined by filter i (0, 1, 2, and 3) in order to determine whether or not a frame is a remote wake-up frame. The MSB (31st bit) must be zero.

The bit j [30:0] is the Byte Mask. If bit j (byte number) of the Byte Mask is set, then the CRC block processes the Filter i Offset + j of the incoming frame; otherwise Filter i Offset + j is ignored.

Filter i Command

The 4-bit filter i command controls the filter i operation.

- Bit 3 (CAST) specifies the address type, defining the destination address type of the pattern.
  When the bit is set, the pattern applies to only multicast packets; when the bit is reset, the pattern applies only to unicast packet.
- Bit 2 (INV), when set, reverses the logic of the CRC16 hash function signal, to reject a packet with matching CRC_16 value.
  Bit 2, along with Bit 1, allows a MAC to reject a subset of remote wake-up packets by creating filter logic such as “Pattern 1 AND NOT Pattern 2”.
- Bit 1 (And_Previous) implements the Boolean logic.
  When set, the result of the current entry is logically ANDed with the result of the previous filter. This AND logic allows a filter pattern longer than 32 bytes by splitting the mask among two, three, or four filters. This depends on the

number of filters that have the And_Previous bit set.
The And_Previous bit is applicable for more than one filter operation, where the filter result is ANDed with the previous filter result. For example, if And_Previous bit is set in Filter 1, the Remote Wakeup packet is detected and PMT interrupt generated only if both Filter 0 and Filter 1 satisfy the Remote Wakeup packet detection and interrupt generation criteria mentioned in **Table 6.135**.

NOTE

- The And_Previous bit setting is applicable within set of 4 filters.
- Setting And_Previous bit of lowest number filter in the set of 4 filters has no effect. For example, setting of And_Previous bit of Filter 0 has no effect.
- If And_Previous bit is set for a filter and the previous filter is not enabled, And_Previous setting has no effect. For example, if Filter 2 And_Previous bit is set (bit 1 of Filter 2 command is set) but Filter 1 is not enabled (bit 0 of in Filter 1 command is reset), then only Filter 1 result is considered.

- Bit 0 (EN) is the enable for filter i. If Bit 0 is not set, filter i is disabled.

Filter i Offset

This Filter i Offset register defines the offset (within the frame) from which the filter i examines the frames. This 8 bits pattern offset is the offset for the filter i first byte to be examined. The minimum allowed offset is 12, which refers to the 13th byte of the frame. The offset value 0 refers to the first byte of the frame.

Filter i CRC 16

This Filter i CRC 16 register contains the CRC_16 value calculated from the pattern and the byte mask programmed to the wake-up filter register block.

The 16-bit CRC calculation uses the following polynomial:

$$G(x) = x^{16} + x^{15} + x^{2} + 1$$

Each mask, used in the hash function calculation, is compared with a 16-bit value associated with that mask. Each filter has the following:

- 32-bit Mask: Each bit in this mask corresponds to one byte in the detected packet. If the bit is 1’, the corresponding byte is taken into the CRC16 calculation.
- 8-bit Offset Pointer: Specifies the byte to start the CRC16 computation.

The pointer and the mask are used together to locate the bytes to be used in the CRC16 calculations.

Table 6.135 lists the Remote Wakeup scenarios in which PMT interrupt is generated

Table 6.135 Remote Wakeup Packet and PMT Interrupt Generation

Filter i Command			Frame Type and CRC Status		Interrupt Generation
CAST	INV	EN	Received Frame Cast Type	CRC Status	RWK INTR
0	0	1	Unicast	MATCH	Remote Wakeup packet is detected and PMT interrupt is generated
0	1	1	Unicast	MISMATCH	Remote Wakeup packet is detected and PMT interrupt is generated
1	0	1	Multicast	MATCH	Remote Wakeup packet is detected and PMT interrupt is generated
1	1	1	Multicast	MISMATCH	Remote Wakeup packet is detected and PMT interrupt is generated

Note: In all other combinations, the Remote Wakeup packet is not detected and PMT interrupt is not generated.

6.6.6 DMA Register

This section defines the bits for each DMA register. The write data input to the DMA registers are qualified with the corresponding byte enable signal inputs (MCI interface). Thus, non 32-bit accesses are allowed as long as the address is Word aligned. Byte, half word, or word accesses are possible.

DMA Registers are in 4400 1000h to 4400 1058h (GMAC1) and 4400 3000h to 4400 3058h (GMAC2). See **Section 6.3, Register Map**.

6.6.6.1 Bus Mode Register (Bus_Mode)

The Bus Mode register establishes the bus operating modes for the DMA.

6.6.6.2 Transmit Poll Demand Register (Transmit_Poll_Demand)

The Transmit Poll Demand register enables the Tx DMA to check whether or not the DMA owns the current descriptor. The Transmit Poll Demand command is given to wake up the Tx DMA if it is in the Suspend mode. The Tx DMA can go into the Suspend mode because of an Underflow error in a transmitted frame or the unavailability of descriptors owned by it. You can give this command anytime, and the Tx DMA resets this command when it again starts fetching the current descriptor from host memory. When this register is read, it always returns zero.

6.6.6.3 Receive Poll Demand Register (Receive_Poll_Demand)

The Receive Poll Demand register enables the Rx DMA to check for new descriptors. This command is given to wake up the Rx DMA from the SUSPEND state. The Rx DMA can go into the SUSPEND state only because of the unavailability of descriptors it owns. When this register is read, it always returns zero.

6.6.6.4 Receive Descriptor List Address Register (Receive_Descriptor_List_Address)

The Receive Descriptor List Address register points to the start of the Receive Descriptor List. The descriptor lists reside in the host’s physical memory space and must be Word aligned (for 32-bit data bus). The DMA internally converts it to bus width aligned address by making the corresponding LS bits low. Writing to this register is permitted only when reception is stopped. When stopped, this register must be written to before the receive Start command is given.

You can write to this register only when Rx DMA has stopped, that is, Bit 1 (SR) is set to zero in register Operation Mode Register (Operation_Mode). When stopped, this register can be written with a new descriptor list address. When you set the SR bit to 1, the DMA takes the newly programmed descriptor base address.

If this register is not changed when the SR bit is set to 0, then the DMA takes the descriptor address where it was stopped earlier.

6.6.6.5 Transmit Descriptor List Address Register (Transmit_Descriptor_List_Address)

The Transmit Descriptor List Address register points to the start of the Transmit Descriptor List. The descriptor lists reside in the host’s physical memory space and must be Word aligned (for 32-bit data bus). The DMA internally converts it to bus width aligned address by making the corresponding LSB to low.

You can write to this register only when the Tx DMA has stopped, that is, Bit 13 (ST) is set to zero in Operation Mode Register (Operation_Mode). When stopped, this register can be written with a new descriptor list address.

When you set the ST bit to 1, the DMA takes the newly programmed descriptor base address.

If this register is not changed when the ST bit is set to 0, then the DMA takes the descriptor address where it was stopped earlier.

6.6.6.6 Status Register (Status)

The Status register contains all status bits that the DMA reports to the host. The Software driver reads this register during an interrupt service routine or polling. Most of the fields in this register cause the host to be interrupted. The bits of this register are not cleared when read. Writing 1’b1 to (unreserved) Bits [16:0] of this register clears these bits and writing 1’b0 has no effect. Each field (Bits[16:0]) can be masked by masking the appropriate bit in Interrupt Enable Register (Interrupt_Enable).

6.6.6.7 Operation Mode Register (Operation_Mode)

The Operation Mode register establishes the Transmit and Receive operating modes and commands. This register should be the last CSR to be written as part of the DMA initialization.

6.6.6.8 Interrupt Enable Register (Interrupt_Enable)

The Interrupt Enable register enables the interrupts reported by Status Register (Status). Setting a bit to 1’b1 enables a corresponding interrupt. After a hardware reset or software reset, all interrupts are disabled.

6.6.6.9 Missed Frame and Buffer Overflow Counter Register (Missed_Frame_And_Buffer_Overflow_Counter)

The DMA maintains two counters to track the number of frames missed during reception. This register reports the current value of the counter. The counter is used for diagnostic purposes. Bits[15:0] indicate missed frames because of the host buffer being unavailable. Bits[27:17] indicate missed frames because of buffer overflow conditions (MTL and MAC) and runt frames (good frames of less than 64 bytes) dropped by the MTL.

6.6.6.10 Receive Interrupt Watchdog Timer Register (Receive_Interrupt_Watchdog_Timer)

This register, when written with a non zero value, enables the watchdog timer for the Receive Interrupt (Bit 6) of Status Register (Status).

6.6.6.11 AXI Bus Mode Register (AXI_Bus_Mode)

The AXI Bus Mode Register controls the behavior of the AXI master. It is mainly used to control the burst splitting and the number of outstanding requests.

6.6.6.12 AXI Status Register (AXI_Status)

This register provides the active status of the read and write channels of the AXI master interface. This register is useful for debugging purposes.

6.6.6.13 Current Host Transmit Descriptor Register (Current_Host_Transmit_Descriptor)

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

6.6.6.14 Current Host Receive Descriptor Register (Current_Host_Receive_Descriptor)

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

6.6.6.15 Current Host Transmit Buffer Address Register (Current_Host_Transmit_Buffer_Address)

The Current Host Transmit Buffer Address register points to the current Transmit Buffer Address being read by the DMA.

6.6.6.16 Current Host Receive Buffer Address Register (Current_Host_Receive_Buffer_Address)

The Current Host Receive Buffer Address register points to the current Receive Buffer address being read by the DMA.

6.6.6.17 HW Feature Register (HW_Feature)

This register indicates the presence of the optional features or functions of the GMAC. The software driver can use this register to dynamically enable or disable the programs related to the optional blocks.

6.6.7 MAC Register

MAC Registers are in 4400 0000h to 4400 080Ch (GMAC1) and 4400 2000h to 4400 280Ch (GMAC2). See **Section 6.3, Register Map**.

6.6.7.1 MAC Configuration Register (MAC_Configuration)

The MAC Configuration register establishes receive and transmit operating modes.

Next table shows how the settings of Bit 27 and Bit 20 of MAC Configuration Register (MAC_Configuration) impact the giant frame status.

Table 6.136 Giant Frame Status based on Bit 27 and Bit 20

Length / Type Field	Received Frame Length	Bit 27 (TWOKPE)	Bit 20 (JE)	Giant Frame Status
Untagged packet	> 1,518	0	0	1
	> 2,000	1	0	1
	> 9,018	x	1	1
VLAN tagged packet	> 1,522	0	0	1
	> 2,000	1	0	1
	> 9,022	x	1	1

Note: For all other combinations, the Giant Frame status is 0.

Next table shows how the settings of Bit 7 and Bit 25 of MAC Configuration Register (MAC Configuration) impact whether CRC length is included in the frame length.

CAUTION

If Type/Length Field < 1,536, the CST field has no effect. Similarly, if Type/Length field > 1,536, the ACS field has no effect.

Table 6.137 Frame Length Based on Bit 7 and Bit 25

Receive Checksum Offload Engine	Received Frame Length	Bit 7 (ACS)	Bit 25 (CST)	FCS Stripping Done
COE Type 2	< 1,536	0	x	No
		1	x	Yes (for Ethernet Frames)
	≥ 1,536	x	0	No
		x	1	Yes (for Type Frames)

6.6.7.2 MAC Frame Filter Register (MAC_Frame_Filter)

The MAC Frame Filter 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.

6.6.7.3 GMII Address Register (GMII_Address)

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

6.6.7.4 GMII Data Register (GMII_Data)

The GMII Data register stores Write data to be written to the PHY register located at the address specified in GMII Address Register (GMII_Address). This register also stores the Read data from the PHY register located at the address specified by GMII_Address.

6.6.7.5 Flow Control Register (Flow_Control)

The Flow Control register controls the generation and reception of the Control (Pause Command) frames by the GMAC's Flow control module. A Write to a register with the Busy bit set to "1" triggers the Flow Control block to generate a Pause frame. The fields of the control frame are selected as specified in the IEEE 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.

6.6.7.6 VLAN Tag Register (VLAN_Tag)

The VLAN Tag register contains the IEEE 802.1Q VLAN Tag to identify the VLAN frames. The GMAC compares the 13th and 14th bytes of the receiving frame (Length/Type) with 16'h8100, and the following two bytes are compared with the VLAN tag. If a match occurs, the GMAC sets the received VLAN bit in the receive frame status. The legal length of the frame is increased from 1,518 bytes to 1,522 bytes.

Consecutive writes to these registers should be performed only after at least four clock cycles in the destination clock domain.

6.6.7.7 Version Register (Version)

The Version registers identifies the version of the GMAC.

6.6.7.8 Debug Register (Debug)

The Debug register gives the status of all main modules of the transmit and receive data paths and the FIFOs. An all zero status indicates that the GMAC is in idle state (and FIFOs are empty) and no activity is going on in the data paths.

CAUTION

The reset values given for the Debug register are valid only if clocks are present during reset operation.

6.6.7.9 LPI Control and Status Register (LPI_Control_Status)

The LPI Control and Status Register controls the LPI functions and provides the LPI interrupt status. The status bits are cleared when this register is read.

6.6.7.10 LPI Timers Control Register (LPI_Timers_Control)

The LPI Timers Control register controls the timeout values in the 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.

6.6.7.11 Interrupt Status Register (Interrupt_Status)

The Interrupt Status register identifies the events in the GMAC that can generate interrupt.

6.6.7.12 Interrupt Mask Register (Interrupt_Mask)

The Interrupt Mask Register bits enable you to mask the interrupt signal because of the corresponding event in the Interrupt Status Register.

6.6.7.13 MAC Address0 High Register (MAC_Address0_High)

The MAC Address0 High register holds the upper 16 bits of the first 6 bytes MAC address of the station. The first DA byte that is received on the (G)MII interface corresponds to the LS byte (Bits [7:0]) of the MAC Address Low register. For example, if 0x112233445566 is received (0x11 in lane 0 of the first column) on the (G)MII as the destination address, then the MacAddress0 Register [47:0] is compared with 0x665544332211.

The synchronization is triggered only when Bits[31:24] of the MAC Address0 Low Register are written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

6.6.7.14 MAC Address0 Low Register (MAC_Address0_Low)

The MAC Address0 Low register holds the lower 32 bits of the 6 bytes first MAC address of the station.

6.6.7.15 MAC Address1 High Register (MAC_Address1_High)

The MAC Address1 High register holds the upper 16 bits of the second 6 bytes MAC address of the station.

The synchronization is triggered only when Bits[31:24] of the MAC Address1 Low Register are written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

6.6.7.16 MAC Address1 Low Register (MAC_Address1_Low)

The MAC Address1 Low register holds the lower 32 bits of the second 6 bytes MAC address of the station.

CAUTION

- The descriptions for MAC Address2 High Register through MAC Address15 High Register are the same as for the MAC Address1 High Register.
- The descriptions for MAC Address2 Low Register through MAC Address15 Low Register are the same as for the MAC Address1 Low Register.
- The descriptions for registers MAC Address16 High Register through MAC Address17 High Register are the same as for the MAC Address1 High Register.
- The descriptions for registers MAC Address16 Low Register through MAC Address17 Low Register are the same as for the MAC Address1 Low Register.

6.6.7.17 Watchdog Timeout Register (WDog_Timeout)

This register controls the watchdog timeout for received frames.

6.6.7.18 Hash Table Register 0 (Hash_Table_Reg0)

This register contains the first 32 bits of the hash table.

The 256-bit Hash table is used for group address filtering. For hash filtering, the content of the destination address in the incoming frame is passed through the CRC logic and the upper eight bits of the CRC register are used to index the content of the Hash table. The most significant bit determines the register to be used (Hash Table Register X), and the least significant five bits determine the bit within the register. For example, a hash value of 8b'10111111 selects Bit 31 of the Hash Table Register 5.

The hash value of the destination address is calculated in the following way:

1. Calculate the 32-bit CRC for the DA (See *IEEE 802.3, Section 3.2.8* for the steps to calculate)
2. Perform bitwise reversal for the value obtained in Step 1.
3. Take the upper 7 (or 8) bits from the value obtained in Step 2.

If the corresponding bit value of the register is 1'b1, the frame is accepted. Otherwise, it is rejected. If the Bit 1 (Pass All Multicast) is set in MAC Frame Filter Register (MAC_Frame_Filter), then all multicast frames are accepted regardless of the multicast hash values.

The synchronization is triggered only when Bits[31:24] of the Hash Table Register X registers are written.

CAUTION

- Consecutive writes to this register should be performed after at least four clock cycles in the destination clock domain.
- Hash Table Register 1 through 7 are similar to Hash Table Register 0.

6.6.7.19 VLAN Hash Table Register (VLAN_Hash_Table_Reg)

The 16-bit Hash table is used for group address filtering based on VLAN tag when Bit 19 (VTHM) of VLAN Tag Register (VLAN_Tag) is set. For hash filtering, the content of the 16-bit VLAN tag or 12-bit VLAN Identifier (based on Bit 16 (ETV) of VLAN Tag Register) in the incoming frame is passed through the CRC logic and the upper four bits of the calculated CRC are used to index the contents of the VLAN Hash table. For example, a hash value of 4b'1000 selects Bit 8 of the VLAN Hash table.

The hash value of the destination address is calculated in the following way:

1. Calculate the 32-bit CRC for the VLAN tag or ID (See IEEE 802.3, Section 3.2.8 for the steps to calculate CRC32).
2. Perform bitwise reversal for the value obtained in Step 1.
3. Take the upper four bits from the value obtained in Step 2.

If the corresponding bit value of the register is 1'b1, the frame is accepted. Otherwise, it is rejected. The synchronization is triggered only when Bits[15:8] of this register are written.

CAUTION

Consecutive writes to this register should be performed after at least four clock cycles in the destination clock domain.

6.6.8 IEEE 1588 Timestamp Registers

This section describes the updated registers required to support the IEEE 1588 functions.

IEEE 1588 Timestamp Registers are in 4400 0700h to 4400 078Ch (GMAC1) and 4400 2700h to 4400 2734h (GMAC2). See **Section 6.3, Register Map**.

6.6.8.1 Timestamp Control Register (Timestamp_Control)

This register controls the operation of the System Time generator and the processing of PTP packets for timestamping in the Receiver.

CAUTION

- Bits[5:1] are reserved when External Timestamp Input feature is enabled (GMAC2).
- Following table indicates the PTP messages, for which a snapshot is taken depending on Bits [17:14] (SNAPTYPSEL), in Timestamp Control Register (Timestamp_Control).

Table 6.138 Timestamp Snapshot Dependency on Register Bits

SNAPTYPSEL (Bits 17:16)	TSMSTRENA (Bit 15)	TSEVNTENA (Bit 14)	PTP Messages
2’b00	X	0	SYNC, Follow_Up, Delay_Req, Delay_Resp
2’b00	0	1	SYNC
2’b00	1	1	Delay_Req
2’b01	X	0	SYNC, Follow_Up, Delay_Req, Delay_Resp, Pdelay_Req, Pdelay_Resp, Pdelay_Resp_Follow_Up
2’b01	0	1	SYNC, Pdelay_Req, Pdelay_Resp
2’b01	1	1	Delay_Req, Pdelay_Req, Pdelay_Resp
2’b10	X	X	SYNC, Delay_Req
2’b11	X	X	Pdelay_Req, Pdelay_Resp

6.6.8.2 Sub Second Increment Register (Sub_Second_Increment)

This register is present only when the IEEE 1588 timestamp feature is selected without an external timestamp input. In the Coarse Update mode (TSCFUPDT bit in Timestamp Control Register (Timestamp_Control)), the value in this register is added to the system time every clock cycle of GMAC_PTP_REFCLK_I. In the Fine Update mode, the value in this register is added to the system time whenever the Accumulator gets an overflow.

This register is present only in GMAC1.

6.6.8.3 System Time – Seconds Register, Nanoseconds Register (System_Time_Seconds, System_Time_Nanoseconds)

The System Time Seconds register, along with System Time Nanoseconds register, indicates the current value of the system time maintained by the GMAC. Though it is updated on a continuous basis, there is some delay from the actual time because of clock domain transfer latencies (from GMAC_PTP_REFCLK_I to CSR clock (GMAC[m]_HCLK)).

These registers (System Time – Seconds and System Time – Nanoseconds) are present only when the IEEE 1588 Timestamp feature is selected without external timestamp input.

This register is present only in GMAC1.

6.6.8.4 System Time – Seconds Update Register, Nanoseconds Update Register (System_Time_Seconds_Update, System_Time_Nanoseconds_Update)

The System Time Seconds Update register, along with the System Time Nanoseconds Update register, initializes or updates the system time maintained by the GMAC. You must write both of these registers before setting the TSINIT or TSUPDT bits in the Timestamp Control register.

This register is present only in GMAC1.

6.6.8.5 Timestamp Addend Register (Timestamp_Addend)

This register is present only when the IEEE 1588 Timestamp feature is selected without external timestamp input. This register value is used only when the system time is configured for Fine Update mode (TSCFUPDT bit in Timestamp Control Register (Timestamp_Control)). This register content is added to a 32-bit accumulator in every clock cycle (of GMAC_PTP_REFCLK_I) and the system time is updated whenever the accumulator overflows.

This register is present only in GMAC1.

6.6.8.6 Target Time Seconds Register, Target Time Nanoseconds Register (Target_Time_Seconds, Target_Time_Nanoseconds)

The Target Time Seconds register, along with Target Time Nanoseconds register, is used to schedule an interrupt event (Timestamp_Status[1] when Advanced Timestamping is enabled; otherwise, TSIS interrupt bit in Interrupt_Status[9]) when the system time exceeds the value programmed in these registers.

This register is present only in GMAC1.

6.6.8.7 Timestamp Status Register (Timestamp_Status)

All bits except Bits[27:25] gets cleared when the host reads this register.

6.6.8.8 PPS Control Register (PPS_Control)

This register is present only in GMAC1.

6.6.8.9 Auxiliary Timestamp – Nanoseconds Register (Auxiliary_Timestamp_Nanoseconds)

This register, along with Auxiliary Timestamp – Seconds Register (Auxiliary_Timestamp_Seconds), gives the 64-bit timestamp stored as auxiliary snapshot. The two registers together form the read port of a 64-bit width FIFO with a depth of 4. Multiple snapshots can be stored in this FIFO. The ATSNS bits in the Timestamp Status register indicate the fill level of this FIFO. The top of the FIFO is removed only when the last byte of Auxiliary Timestamp – Seconds Register is read. This means when Bits[31:24] are read.

6.6.8.10 PPS0 Interval Register (PPS0_Interval)

The PPS0 Interval register contains the number of units of sub second increment value between the rising edges of PPS0 signal output (MAC_PPS[0]).

This register is present only in GMAC1.

6.6.8.11 PPS0 Width Register (PPS0_Width)

The PPS0 Width register contains the number of units of sub second increment value between the rising and corresponding falling edges of the PPS0 signal output (MAC_PPS[0]).

This register is present only in GMAC1.

6.6.8.12 PPS1 Target Time Seconds Register (PPS1_Target_Time_Seconds)

The PPS1 Target Time Seconds register, along with PPS1 Target Time Nanoseconds register, is used to schedule an interrupt event (Bit 1 (TSTARGT) of Timestamp Status Register (Timestamp_Status)) when the system time exceeds the value programmed in these registers.

This register is present only in GMAC1.

6.6.8.13 PPS1 Target Time Nanoseconds Register (PPS1_Target_Time_Nanoseconds)

This register is present only in GMAC1.

CAUTION

- PPS1 Interval Register (PPS1_Interval) is similar to PPS0 Interval Register (PPS0_Interval).
- PPS1 Width Register (PPS1_Width) is similar to PPS0 Width Register (PPS0_Width).

Section 7 HSR Switch

7.1 Overview

- HSR functionality (IEC 62439-3 edition 2.0 - 2012)
  - DANH: 2 ports used
  - Redundancy Box (Red Box): 3 ports used (one port to single attached node)
  - Generation of redundant transmit frames
  - Filtering of duplicated received frames
  - Redundancy header generation and detection
  - Table to keep track of received frames
- 100 Mbps full-duplex Ethernet only supported
- Dynamic frame buffer allocation (page manager)
- 128 proxy nodes (VDANs) supported
- CPU interface for receiving and sending frames
- Support for link local protocols – CPU may send to specific ports only – CPU knows receive port
- Duplicate detection memory
- MAC address filtering (8 filter masks for interlink port, 6 for CPU)
- One VLAN tag supported
- Port statistics on per port basis (no aggregation)
  - Access to a subset of MIB through SNMP via the Host port
- Frame buffer size: 144 KByte
  - Statically shared memory with HW-RTOS GMAC (HSR feature and HW-RTOS GMAC feature are exclusive).
- Support of IEEE1588 – 2008:
  - PTPv2 timestamping on Port A, B, Interlink
  - Ordinary clock (OC) / boundary clock (BC)
  - Transparent clock (TC) between port A and B
  - MII pass through (allows one step or two step operation)
  - Low latency (960 ns at 100 Mbps)
  - Independent timestamping for BC/OC and TC clock
  - Recognized headers in PTP frames: HSR, VLAN
- PTP interface
  - 32 bits input (current second value)
  - 30 bits input (current nanosecond value)
  - PTP clock for signals re-synchronization.
- Support for Ethernet multicast frames with flooding control
- Extended frame size: Up to 2000 Bytes (Jumbo frames not supported)
- Interface:

- Native mode MII
- No Native RMII, managed by RMII/RGMII convertor (connected on external pins)

- Support for a minimum of 16 nodes in an HSR loop.
- Target is 10 nodes with 65 Mbits bandwidth in HSR loop.

Figure 7.1 HSR Block Diagram

7.1.1 HSR RedBox

In HSR-SAN Mode, the HSR Core receives standard Ethernet frames from the interlink port. It adds a HSR header and sends a copy to both ring ports. Frames from the ring are forwarded to the interlink port without HSR header. Duplicates are discarded using the HSR sequence number in combination with the source address of the frame.

In HSR-HSR Mode it can be used to connect two HSR rings. Two RedBoxes from different rings are connected through their interlink port to form a QuadBox. Two QuadBoxes are used to make the transition redundant.

Apart from RedBoxes, a HSR ring may also include doubly attached nodes with native HSR support (DANH). These nodes support only one source MAC address behind the interlink port.

Figure 7.2 HSR Ring and Node Types

7.2 Signal Interface

Signal Name	I/O	Description	Active
Clock			
HSR_HCLK	I	AHB clock	
HSR_CLK100	I	100 MHz clock	
HSR_CLK50	I	50 MHz clock	
Signals for PTP			
HSR_OVERWRITE_TIME_I	I	Timestamp value for PTP	
Interrupt			
HSR_CPU_I_IRQ	O	CPU interface interrupt, level sensitive	High
HSR_PTP_I_IRQ	O	PTP interface interrupt, level sensitive	High

7.3 Register Map

Table 7.1 HSR Register Map (1/2)

Address	Register Symbol	Register Name
4404 0000h	RCI_WCONFIG	Send Config Register
4404 0004h	RCI_RCONFIG	Receive Config Register
4404 0008h	RCI_INT	Interrupt Register
4404 000Ch + 4h × n	RCI_TXRX_DATA[n] (n = 0..508)	Frame Data Send Register/Frame Data Receive Register
4404 8000h	R_MACADL	Own MAC Address Low Register
4404 8004h	R_MACADH	Own MAC Address High Register
4404 8008h	R_TST1	Alive Test 1 Register
4404 800Ch	R_TST2	Alive Test 2 Register
4404 8010h	R_PEN	Port Enable Register
4404 8014h	R_PNT_AGT	Proxynode Table Aging Time Register
4404 8018h	R_DD_AGT	Duplicate detection Aging Time Register
4404 801Ch + 8h × (m − 1)	R_MACFLT_I[m]L (m = 1 to 8)	Filter MAC Address Interlink [m] Low Register
4404 8020h + 8h × (m − 1)	R_MACFLT_I[m]H (m =1 to 8)	Filter MAC Address Interlink [m] High Register
4404 805Ch + 8h × (m − 1)	R_MACFLT_C[m]L (m =1 to 6)	Filter MAC Address CPU [m] Low Register
4404 8060h + 8h × (m − 1)	R_MACFLT_C[m]H (m =1 to 6)	Filter MAC Address CPU [m] High Register
4404 808Ch	R_VER	HSR Version Register
4404 8090h	R_RAM_STA	RAM Status Register
4404 8094h	R_UFMC	Used Frame Memory Count Register
4404 8098h	R_FRA_ALL_ARX	Received Frame A Register
4404 809Ch	R_FRA_TAG_ARX	Received tagged Frame A Register
4404 80A0h	R_FRA_NLL_ARX	Received not linklocal frame A Register
4404 80A4h	R_FRA_ERR_ARX	Received Error Frame A Register
4404 80A8h	R_FRA_WRO_ARX	Wrong LAN Frame A Register
4404 80ACh	R_FRA_ALL_ATX	Sent Frame A Register
4404 80B0h	R_FRA_TAG_ATX	Sent tagged Frame A Register
4404 80B4h	R_FRA_NLL_ATX	Sent not linklocal Frame A Register
4404 80B8h	R_FRA_ALL_BRX	Received Frame B Register
4404 80BCh	R_FRA_TAG_BRX	Received tagged Frame B Register
4404 80C0h	R_FRA_NLL_BRX	Received not linklocal frame B Register
4404 80C4h	R_FRA_ERR_BRX	Received Error B Register
4404 80C8h	R_FRA_WRO_BRX	Wrong LAN Count B Register
4404 80CCh	R_FRA_ALL_BTX	Sent Frame B Register
4404 80D0h	R_FRA_TAG_BTX	Sent tagged Frame B Register
4404 80D4h	R_FRA_NLL_BTX	Sent not linklocal Frame B Register
4404 80D8h	R_FRA_ALL_CRX	Received Frame C Register
4404 80DCh	R_FRA_TAG_CRX	Received tagged Frame C Register
4404 80E0h	R_FRA_NLL_CRX	Received not linklocal frame C Register
4404 80E4h	R_FRA_ERR_CRX	Received Error C Register
4404 80E8h	R_FRA_WRO_CRX	Wrong LAN Count C Register
4404 80ECh	R_FRA_ALL_CTX	Sent Frame C Register
4404 80F0h	R_FRA_TAG_CTX	Sent tagged Frame C Register
4404 80F4h	R_FRA_NLL_CTX	Sent not linklocal Frame C Reg
4404 80F8h	R_FREE_FRA_M	Free Frame Memory Count
4404 80FCh	R_DBG_RPT1	Internal Debug Reports Register 1

Table 7.1 HSR Register Map (2/2)

Address	Register Symbol	Register Name
4404 8100h	R_DBG_RPT2	Internal Debug Reports Register 2
4404 8104h	R_PNT_S	Proxynodetable status Register
4404 8108h	R_PNT_D	Proxynodetable data Register
4404 C000h	RPTP_ID	PTP core ID Register
4404 C004h	RPTP_TST	PTP core Test Register
4404 C008h	RPTP_VER	PTP Version Register
4404 C00Ch	RPTP_GPO	General purpose Register
4404 C014h	RPTP_INT	PTP Interrupt Register
4404 C018h	RPTP_INT_MSK	PTP Interrupt Mask Register
4404 C01Ch to 4404 C040h	(Reserved)	
4404 C044h + 18h × (m − 1)	RPTP_TS_STAT_[m] (m = 1 to 4)	Timestamp status Register P[m]
4404 C048h + 18h × (m − 1)	RPTP_TS_RD_[m] (m = 1 to 4)	Timestamp read Register P[m]
4404 C04Ch + 18h × (m − 1)	RPTP_PORT_CONF_[m] (m = 1 to 4)	Port Config Register P[m]
4404 C050h + 18h × (m − 1)	RPTP_P_DELAY_[m] (m = 1 to 4)	Peer Delay Register P[m]
4404 C054h + 18h × (m − 1)	RPTP_PHY_DLY_TX_[m] (m = 1 to 4)	PHY Tx delay Register P[m]
4404 C058h + 18h × (m − 1)	RPTP_PHY_DLY_RX_[m] (m = 1 to 4)	PHY Rx delay Register P[m]
4404 C0A4h	RPTP_BUF_STAT	Buffer Status Register

7.4 Register Description

7.4.1 RCI_WCONFIG — Send Config Register

The bits of this register should be written at once.

Address: 4404 0000h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	SB	—	—	END	—	—	—	—	PTH				TAG	B	A	I
Value after reset	0	X	X	0	X	X	X	X	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	WFS											
Value after reset	X	X	X	X	0	0	0	0	0	0	0	0	0	0	0	0

Table 7.2 RCI_WCONFIG Register Contents

Bit Position	Bit Name	Function	R/W
b31	SB	SEND BIT Start sending the Frame if set to one. This bit will automatically be set to zero after the frame was sent.	W
b30, b29	Reserved	Set to zero on Write. ignore on Read.	R
b28	END	ENDIAN SWAP When enabled, frame data written will be interpreted as little-endian (the least significant byte will be the first on the wire).	W
b27 to b24	Reserved	Set to zero on Write. ignore on Read.	R
b23 to b20	PTH	PATH Path field of the HSR tag.	W
b19	TAG	SEND WITH TAG If set, the frame will be sent with tag and sequence number on port A/B. Otherwise it will be sent without tag.	W
b18	B	SEND TO PORT B Send the frame to Port B, respecting the TAG bit.	W
b17	A	SEND TO PORT A Send the frame to Port A, respecting the TAG bit.	W
b16	I	SEND TO INTERLINK Send the frame to Interlink, usually without tag, ignoring the TAG bit. For the HSR RedBox in HSR-HSR or HSR-PRP mode the TAG bit is respected.	W
b15 to b12	Reserved	Set to zero on Write. ignore on Read.	R
b11 to b0	WFS	WRITE FRAME SIZE Size of the send Frame without preamble, SFD and FCS	W

7.4.2 RCI_RCONFIG — Receive Config Register

Address: 4404 0004h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	RFD	FP	—	END	—	—	—	—	—	—	—	—	TAG		RPT	
Value after reset	0	0	X	0	X	X	X	X	X	X	X	X	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	RFS											
Value after reset	X	X	X	X	0	0	0	0	0	0	0	0	0	0	0	0

Table 7.3 RCI_RCONFIG Register Contents

Bit Position	Bit Name	Function	R/W
b31	RFD	READ FRAME DONE Handshake bit from the CPU to the Buffer. The Hardware will free the frame buffer, when set to 1.	W
b30	FP	FRAME PENDING 0: no more frames 1: received frame is in the buffer Should be read after each frame read, to check if a new frame was received while reading another frame to the CPU. So a new read will be required.	R
b29	Reserved	Set to zero on Write. ignore on Read.	R
b28	END	ENDIAN SWAP When enabled, frame will be read as little-endian (the least significant byte was the first on wire)	R/W
b27 to b20	Reserved	Set to zero on Write. ignore on Read.	R
b19, b18	TAG	RECEIVED WITH TAG 00b: no tag detected at RX port 01b: HSR header 10b: PRP-1 trailer 11b: Reserved	R
b17, b16	RPT	RECEIVE PORT 00b: Reserved 01b: Interlink 10b: Port A 11b: Port B	R
b15 to b12	Reserved	Set to zero on Write. ignore on Read.	R
b11 to b0	RFS	READ FRAME SIZE Size of the received Frame without PREAMBLE, SFD and FCS	R

7.4.3 RCI_INT — Interrupt Register

Address: 4404 0008h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TXI	RXI
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	0	0

Table 7.4 RCI_INT Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b2	Reserved	Set to zero on Write. ignore on Read.	R
b1	TXI	TX INTERRUPT Interrupt will be 1 if a frame is completely send to the switch. Bit will be cleared on reading	R
b0	RXI	RX INTERRUPT Interrupt will be 1 if a frame is completely received from the switch. Bit will be cleared on reading	R

7.4.4 RCI_TXRX_DATA[n] — Frame Data Send / Receive Register (n = 0..508)

Address: 4404 000Ch + 4h × n

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	BYTE4								BYTE3							
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	BYTE2								BYTE1							
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Table 7.5 RCI_TXRX_DATA[n] Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b24	BYTE4	BYTE[4+4 × n] Read: Receive data Byte*1 Write: Send data Byte*2	R/W
b23 to b16	BYTE3	BYTE[3+4 × n] Read: Receive data Byte*1 Write: Send data Byte*2	R/W
b15 to b8	BYTE2	BYTE[2+4 × n] Read: Receive data Byte*1 Write: Send data Byte*2	R/W
b7 to b0	BYTE1	BYTE[1+4 × n] Read: Receive data Byte*1 Write: Send data Byte*2	R/W

Note 1. PREAMBLE, SFD and FCS are not stored in this register. Valid byte lane are BYTE[m] (m = 1,2,3,…,RFS). BYTE[k] (k = RFS+1, RFS+2,…) are invalid.

Note 2. Do not write PREAMBLE, SFD and FCS to this register. Valid byte lane are BYTE[m] (m = 1,2,3,…,WFS). BYTE[k] (k = WFS+1, WFS+2,…) are invalid.

7.4.5 R_MACADL — Own MAC Address Low Register

Address: 4404 8000h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	MAL															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MAL															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 7.6 R_MACADL Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	MAL	MAC ADDRESS LOW First four Byte of the Device MAC Address (Little Endian)	R/W

7.4.6 R_MACADH — Own MAC Address High Register

Address: 4404 8004h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	MLE	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MAH															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 7.7 R_MACADH Register Contents

Bit Position	Bit Name	Function	R/W
b31	MLE	MAC LEARNING ENABLE Enables the learning of the Device MAC Address. Not available in RZ/N1.	R/W
b30 to b16	Reserved	Set to zero on Write. ignore on Read.	R
b15 to b0	MAH	MAC ADDRESS HIGH: The two highest Byte of the Device MAC Address Little Endian	R/W

7.4.7 R_TST1 — Alive Test 1 Register

Address: 4404 8008h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TS1															
Value after reset	1	0	1	0	1	1	0	0	1	1	0	0	1	1	1	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TS1															
Value after reset	0	1	0	1	0	1	0	1	1	1	1	0	1	1	0	1

Table 7.8 R_TST1 Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	TS1	TEST 1 Reading this register returns the value 0xACCE55ED if the device is up and running	R

7.4.8 R_TST2 — Alive Test 2 Register

Address: 4404 800Ch

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TS2															
Value after reset	1	0	1	0	1	1	0	0	1	1	0	0	1	1	1	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TS2															
Value after reset	0	1	0	1	0	1	0	1	1	1	1	0	1	1	0	1

Table 7.9 R_TST2 Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	TS2	TEST 2 Write any value to this register. It will be returned XORed by the value 0xACCE55ED on the next read.	R/W

7.4.9 R_PEN — Port Enable Register

Address: 4404 8010h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	RBM			HPC				HPI				NOM			—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	CUP	BTE	ATE	ITE	CTE	BRE	ARE	IRE	CRE
Value after reset	X	X	X	X	X	X	X	1	0	0	0	0	0	0	0	0

Table 7.10 R_PEN Register Contents (1/2)

Bit Position	Bit Name	Function	R/W
b31 to b29	RBM	RedBox Operation Mode 000b: HSR-SAN Mode 001b: HSR-PRP A Mode 010b: HSR-PRP B Mode 011b: HSR-HSR Mode	R/W
b28 to b25	HPC	HSR Path CPU Path field to set in the HSR tag for frames received from the CPU interface	R/W
b24 to b21	HPI	HSR Path Interlink Path field to set in the HSR tag for frames received from the Interlink port. The 3 most significant bits of this are known as “network identity” in the HSR standard. If the RedBox is in PRP A Mode or PRP B Mode, frames matching this network identity will not be forwarded from the HSR ring ports to the interlink.	R/W
b20 to b18	NOM	Node Operating Mode 000b: forwarding with HSR tag (mode H) 001b: no forwarding (mode N) 010b: forwarding transparently all (mode T) 100b: forwarding unicast (mode U)	R/W
b17 to b9	Reserved	Set to zero on Write. ignore on Read.	R
b8	CUP	Cut PRP Trailer 1: Cut PRP trailer 0: Don’t Cut PRP trailer	R/W
b7	BTE	Port B TX Enable 1: Sending Frame Enable 0: Disable	R/W
b6	ATE	Port A TX Enable 1: Sending Frame Enable 0: Disable	R/W
b5	ITE	Interlink TX Enable 1: Sending Frame Enable 0: Disable	R/W
b4	CTE	CPU TX Enable 1: Sending Frame Enable 0: Disable	R/W
b3	BRE	Port B RX Enable 1: Receiving Frame Enable 0: Disable	R/W

Table 7.10 R_PEN Register Contents (2/2)

Bit Position	Bit Name	Function	R/W
b2	ARE	Port A RX Enable 1: Receiving Frame Enable 0: Disable	R/W
b1	IRE	Interlink RX Enable 1: Receiving Frame Enable 0: Disable	R/W
b0	CRE	CPU RX Enable 1: Receiving Frame Enable 0: Disable	R/W

7.4.10 R_PNT_AGT — Proxynode Table Aging Time Register

Address: 4404 8014h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	AT															
Value after reset	0	0	0	0	1	0	1	1	0	0	1	0	1	1	0	1

Table 7.11 R_PNT_AGT Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b16	Reserved	Set to zero on Write. ignore on Read.	R
b15 to b0	AT	AGING TIME Aging time of Proxynode Table. The default value is 60 seconds. The value for this register is calculated by the following formula: T_aging / (64 × N_proxy_nodes_max^2 × aging_clock_duration) Where T_aging is 60s, N_proxy_nodes_max is 128, aging_clock_duration is 20ns.	R/W

7.4.11 R_DD_AGT — Duplicate Detection Aging Time Register

Address: 4404 8018h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	DIS	LEN				—	—	—	—	—	—	—	—	—	AT	
Value after reset	0	0	1	0	0	X	X	X	X	X	X	X	X	X	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	AT															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Table 7.12 R_DD_AGT Register Contents

Bit Position	Bit Name	Function	R/W
b31	DIS	AGING DISABLE Set this bit to 1 to disable aging of the duplicate table	R/W
b30 to b27	LEN	LENGTH The length of the duplicate detection is 256×LEN.	R
b26 to b18	Reserved	Set to zero on Write. ignore on Read.	R
b17 to b0	AT	AGING TIME The number of clocks after which a dummy duplicate entry is written, overwriting the oldest entry. The minimum value is 2, which clears one entry every 40ns. A new duplicate entry is cleared by aging after approx. 20 ns×AT×256×LEN.	R/W

7.4.12 R_MACFLT_I[m]L — Filter MAC Address Interlink [m] Low Register (m = 1 to 8)

Address: 4404 801Ch + 8h × (m − 1)

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	FML															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	FML															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 7.13 R_MACFLT_I[m]L Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	FML	MAC ADDRESS LOW Lowest four Byte of the first MAC-Filter MAC Address [m] Little Endian Only the frames matched with TX MAC address are forwarded.	R/W

7.4.13 R_MACFLT_I[m]H — Filter MAC Address Interlink [m] High Register (m = 1 to 8)

Address: 4404 8020h + 8h × (m − 1)

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	FBM												—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	FMH															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 7.14 R_MACFLT_I[m]H Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b20	FBM	FILTER BIT MASK Each bit enable one Nibble of the MAC-Filter MAC Address [m]	R/W
b19 to b16	Reserved	Set to zero on Write. ignore on Read.	R
b15 to b0	FMH	MAC ADDRESS HIGH The two highest Byte of the first MAC-Filter MAC Address [m] Little Endian Only the frames matched with TX MAC address are forwarded.	R/W

7.4.14 R_MACFLT_C[m]L — Filter MAC Address CPU [m] Low Register (m = 1 to 6)

Address: 4404 805Ch + 8h × (m − 1)

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	FML															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	FML															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 7.15 R_MACFLT_C[m]L Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	FML	MAC ADDRESS LOW First four Byte of the CPU MAC-Filter MAC Address [m] Little Endian Only the frames matched with TX MAC address are forwarded.	R/W

7.4.15 R_MACFLT_C[m]H — Filter MAC Address CPU [m] High Register (m = 1 to 6)

Address: 4404 8060h + 8h × (m − 1)

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	FBM												—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	FMH															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 7.16 R_MACFLT_C[m]H Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b20	FBM	FILTER BIT MASK Each bit enable one Nibble of the MAC-Filter MAC Address [m]	R/W
b19 to b16	Reserved	Set to zero on Write. ignore on Read.	R
b15 to b0	FMH	MAC ADDRESS HIGH The two highest Byte of the MAC-Filter MAC Address [m] Little Endian Only the frames matched with TX MAC address are forwarded.	R/W

7.4.16 R_VER — HSR Version Register

Address: 4404 808Ch

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	NT		—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	1	0	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CVM								CVS							
Value after reset	0	0	0	0	0	1	0	0	0	1	1	1	0	0	1	0

Table 7.17 R_VER Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b30	NT	NODE TYPE Specifies the Node Type of the HSR Core (2 = HSR, as in the MIB)	R
b29 to b16	Reserved	Set to zero on Write. ignore on Read.	R
b15 to b8	CVM	CORE VERSION MAIN Main Release Number	R
b7 to b0	CVS	CORE VERSION SUB Sub Release Number	R

7.4.17 R_RAM_STA — RAM Status Register

Address: 4404 8090h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	FP															
Value after reset	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	FH															
Value after reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

Table 7.18 R_RAM_STA Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b16	FP	FREE PAGES Number of free data pages	R
b15 to b0	FH	FREE HEADERS Number of free header ram entries	R

7.4.18 R_UFMC — Used Frame Memory Count Register

Address: 4404 8094h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	UF															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	UF															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 7.19 R_UFMC Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	UF	USED FRAME Sum of all Frames written in to the Data RAM Should be the same value then “Free Frame Memory Count Register”.	R

7.4.19 R_FRA_ALL_ARX — Received Frame A Register

Address: 4404 8098h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	ALL															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ALL															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 7.20 R_FRA_ALL_ARX Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	ALL	RECEIVED FRAMES Number of receive started frames on Port A	R

7.4.20 R_FRA_TAG_ARX — Received Tagged Frame A Register

Address: 4404 809Ch

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TAG															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TAG															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 7.21 R_FRA_TAG_ARX Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	TAG	RECEIVED TAGGED FRAMES Number of tagged frames received on Port A	R

7.4.21 R_FRA_NLL_ARX — Received Not linklocal Frame A Register

Address: 4404 80A0h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	NLL															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	NLL															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 7.22 R_FRA_NLL_ARX Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	NLL	RECEIVED NOT LINKLOCAL FRAMES Number of not linklocal frames received on Port A	R

7.4.22 R_FRA_ERR_ARX — Received Error Frame A Register

Address: 4404 80A4h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	ERR															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ERR															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 7.23 R_FRA_ERR_ARX Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	ERR	RECEIVED FRAME ERRORS Number of receive error input (RX_ERR) on Port A	R

7.4.23 R_FRA_WRO_ARX — Wrong LAN Frame A Register

Address: 4404 80A8h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	WRO															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	WRO															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 7.24 R_FRA_WRO_ARX Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	WRO	WRONG LAN FRAME (PRP-1 RedBox only) Number of frames received on Port A with wrong LAN ID in PRP-1 Trailer	R

7.4.24 R_FRA_ALL_ATX — Sent Frame A Register

Address: 4404 80ACh

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	ALL															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ALL															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 7.25 R_FRA_ALL_ATX Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	ALL	SENT FRAMES Number of normally sent frames on Port A	R

7.4.25 R_FRA_TAG_ATX — Sent Tagged Frame A Register

Address: 4404 80B0h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TAG															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TAG															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 7.26 R_FRA_TAG_ATX Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	TAG	SENT TAGGED FRAMES Number of frames sent on Port A when tagged frame received	R

7.4.26 R_FRA_NLL_ATX — Sent Not Linklocal Frame A Register

Address: 4404 80B4h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	NLL															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	NLL															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 7.27 R_FRA_NLL_ATX Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	NLL	SENT FRAMES Number of not linklocal frames sent on Port A Linklocal frames from CPU are also counted.	R

7.4.27 R_FRA_ALL_BRX — Received Frame B Register

Address: 4404 80B8h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	ALL															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ALL															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 7.28 R_FRA_ALL_BRX Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	ALL	RECEIVED FRAMES Number of receive started frames on Port B	R

7.4.28 R_FRA_TAG_BRX — Received Tagged Frame B Register

Address: 4404 80BCh

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TAG															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TAG															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 7.29 R_FRA_TAG_BRX Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	TAG	RECEIVED TAGGED FRAMES Number of tagged frames received on Port B	R

7.4.29 R_FRA_NLL_BRX — Received Not Linklocal Frame B Register

Address: 4404 80C0h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	NLL															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	NLL															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 7.30 R_FRA_NLL_BRX Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	NLL	RECEIVED NOT LINKLOCAL FRAMES Number of not linklocal frames received on Port B	R

7.4.30 R_FRA_ERR_BRX — Received Error B Register

Address: 4404 80C4h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	ERR															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ERR															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 7.31 R_FRA_ERR_BRX Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	ERR	RECEIVED FRAME ERRORS Number of receive error input (RX_ERR) on Port B	R

7.4.31 R_FRA_WRO_BRX — Wrong LAN Count B Register

Address: 4404 80C8h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	WRO															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	WRO															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 7.32 R_FRA_WRO_BRX Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	WRO	WRONG LAN FRAME (PRP-1 RedBox only) Number of frames received on Port B with wrong LAN ID in PRP-1 Trailer	R

7.4.32 R_FRA_ALL_BTX — Sent Frame B Register

Address: 4404 80CCh

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	ALL															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ALL															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 7.33 R_FRA_ALL_BTX Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	ALL	SENT FRAMES Number of normally sent frames on Port B	R

7.4.33 R_FRA_TAG_BTX — Sent Tagged Frame B Register

Address: 4404 80D0h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TAG															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TAG															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 7.34 R_FRA_TAG_BTX Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	TAG	SENT TAGGED FRAMES Number of frames sent on Port B when tagged frame received	R

7.4.34 R_FRA_NLL_BTX — Sent Not Linklocal Frame B Register

Address: 4404 80D4h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	NLL															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	NLL															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 7.35 R_FRA_NLL_BTX Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	NLL	SENT FRAMES Number of not linklocal frames sent on Port B Linklocal frames from CPU are also counted.	R

7.4.35 R_FRA_ALL_CRX — Received Frame C Register

Address: 4404 80D8h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	ALL															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ALL															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 7.36 R_FRA_ALL_CRX Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	ALL	RECEIVED FRAMES Number of receive started frames on Interlink Port	R

7.4.36 R_FRA_TAG_CRX — Received Tagged Frame C Register

Address: 4404 80DCh

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TAG															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TAG															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 7.37 R_FRA_TAG_CRX Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	TAG	RECEIVED TAGGED FRAMES Number of tagged frames received on Interlink Port	R

7.4.37 R_FRA_NLL_CRX — Received Not Linklocal Frame C Register

Address: 4404 80E0h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	NLL															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	NLL															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 7.38 R_FRA_NLL_CRX Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	NLL	RECEIVED NOT LINKLOCAL FRAMES Number of not linklocal frames received on Interlink Port	R

7.4.38 R_FRA_ERR_CRX — Received Error C Register

Address: 4404 80E4h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	ERR															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ERR															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 7.39 R_FRA_ERR_CRX Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	ERR	RECEIVED FRAME ERRORS Number of receive error input (RX_ERR) on Interlink Port	R

7.4.39 R_FRA_WRO_CRX — Wrong LAN Count C Register

Address: 4404 80E8h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	WRO															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	WRO															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 7.40 R_FRA_WRO_CRX Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	WRO	WRONG LAN FRAME (PRP-1 RedBox only) Number of frames received on Interlink Port with wrong LAN ID in PRP-1 Trailer	R

7.4.40 R_FRA_ALL_CTX — Sent Frame C Register

Address: 4404 80ECh

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	ALL															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ALL															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 7.41 R_FRA_ALL_CTX Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	ALL	SENT FRAMES Number of normally sent frames on Interlink Port	R

7.4.41 R_FRA_TAG_CTX — Sent Tagged Frame C Register

Address: 4404 80F0h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TAG															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TAG															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 7.42 R_FRA_TAG_CTX Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	TAG	SENT TAGGED FRAMES Number of frames sent on Interlink Port when tagged frame received	R

7.4.42 R_FRA_NLL_CTX — Sent Not Linklocal Frame C Reg

Address: 4404 80F4h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	NLL															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	NLL															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 7.43 R_FRA_NLL_CTX Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	NLL	SENT FRAMES Number of not linklocal frames sent on Interlink Port Linklocal frames from CPU are also counted	R

7.4.43 R_FREE_FRA_M — Free Frame Memory Count

Address: 4404 80F8h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	FF															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	FF															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 7.44 R_FREE_FRA_M Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	FF	FREED FRAMES Sum of all Freed Frames in the Data RAM Should be the same value then “Used Frame Memory Count Register”	R

7.4.44 R_DBG_RPT1 — Internal Debug Reports Register 1

Address: 4404 80FCh

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	RES															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RES			D44	D43	D42	D41	D40	D39	D38	D37	D36	D35	D34	D33	D32
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 7.45 R_DBG_RPT1 Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b13	RES	Set to zero on Write. ignore on Read.	R/W
b12	D44	INTERNAL ERROR*1	R/W
b11	D43	RX VERY SHORT FRAME*1	R/W
b10	D42	RX SFD MISSING*1	R/W
b9	D41	INTERNAL ERROR*1	R/W
b8	D40	INTERNAL ERROR*1	R/W
b7	D39	INTERNAL ERROR*1	R/W
b6	D38	INTERNAL ERROR*1	R/W
b5	D37	INTERNAL ERROR*1	R/W
b4	D36	INTERNAL ERROR*1	R/W
b3	D35	INTERNAL ERROR*1	R/W
b2	D34	INTERNAL ERROR*1	R/W
b1	D33	RX FREEING FRAME*1	R/W
b0	D32	RX CRC ERROR*1	R/W

Note 1. Cleared by 0 write

7.4.45 R_DBG_RPT2 — Internal Debug Reports Register 2

Address: 4404 8100h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	D31	D30	D29	D28	D27	D26	D25	D24	D23	D22	D21	D20	D19	D18	D17	D16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 7.46 R_DBG_RPT2 Register Contents (1/2)

Bit Position	Bit Name	Function	R/W
b31	D31	RX FRAME ERROR*1 ***Caution*)** RX controller: frame_error reported by PHY	R/W
b30	D30	RX OVERSIZED FRAME*1 ***Caution*)** RX controller: received a frame longer than 2000 bytes	R/W
b29	D29	RX SHORT FRAME*1 ***Caution*)** RX controller: received a frame shorter than 64 bytes	R/W
b28	D28	RX DESTROY_FRAME*1 ***Note*)** RX controller: got destroy_frame command from switch	R/W
b27	D27	RX SWITCH NOT READY*1 ***Note*)** RX controller: switch not ready, discarding a frame	R/W
b26	D26	RX DATA-RAM FULL*1 ***Caution*)** RX controller: Data-RAM full, discarding a frame	R/W
b25	D25	TX TRUNCATING*1 ***Caution*)** TX controller: truncating a frame and generating an invalid FCS	R/W
b24	D24	INTERNAL ERROR*1	R/W
b23	D23	INTERNAL ERROR*1	R/W
b22	D22	INTERNAL ERROR*1	R/W
b21	D21	INTERNAL ERROR*1	R/W
b20	D20	CPU QUEUE FULL*1 ***Caution*)** CPU TX queue full	R/W
b19	D19	PORT A QUEUE FULL*1 ***Caution*)** Port A TX queue full	R/W
b18	D18	PORT B QUEUE FULL*1 ***Caution*)** Port B TX queue full	R/W
b17	D17	INTERLINK QUEUE FULL*1 ***Caution*)** Interlink TX queue full	R/W
b16	D16	RESERVED Set to zero on Write. ignore on Read.	R/W
b15	D15	DATA-RAM FULL*1 ***Caution*)** Data-RAM full	R/W

Table 7.46 R_DBG_RPT2 Register Contents (2/2)

Bit Position	Bit Name	Function	R/W
b14	D14	RESERVED Set to zero on Write. ignore on Read.	R/W
b13	D13	RESERVED Set to zero on Write. ignore on Read.	R/W
b12	D12	INTERNAL ERROR*1	R/W
b11	D11	INTERNAL ERROR*1	R/W
b10	D10	INTERNAL ERROR*1	R/W
b9	D9	HEADER RAM FULL*1 ***Caution*)** Header RAM full	R/W
b8	D8	TX SKIP FRAME*1 ***Note*)** TX controller: skipping a frame (frame was marked to be discarded)	R/W
b7	D7	TX DISCARDING DUPLICATE*1 ***Note*)** Tx controller: discarding a duplicate	R/W
b6	D6	INTERNAL ERROR*1	R/W
b5	D5	READY STILL HIGH*1 ***Note*)** Proxynode table: short frame, address_ready high while still searching	R/W
b4	D4	INTERNAL ERROR*1	R/W
b3	D3	PROXYNODE TABLE FULL*1 ***Caution*)** Proxynode table full, dropping the new node	R/W
b2	D2	NO PROXYNODE ENTRY*1 ***Note*)** No Proxynode table entry because the received frame was already tagged	R/W
b1	D1	INTERNAL ERROR*1	R/W
b0	D0	INTERNAL ERROR*1	R/W

Note 1. Cleared by 0 write

7.4.46 R_PNT_S — Proxynodetable Status Register

Address: 4404 8104h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	PNTS															
Value after reset	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PNTP															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 7.47 R_PNT_S Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b16	PNTS	Proxynodetable size Returns the maximum proxynodetable size	R
b15 to b0	PNTP	Proxynodetable pointer Return which element is in the data register. Increments on every read on the R_PNT_D register. Read-only. To set this to a desired value, you have to read R_PNT_D several times.	R

7.4.47 R_PNT_D — Proxynodetable Data Register

Address: 4404 8108h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	DATA															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DATA															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 7.48 R_PNT_D Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	DATA	Proxynodetable data Contains 32bit of the proxynodetable where the PNTP field of R_PNT_S register points to. The PNTP field is incremented by one after every read of this register. A proxynodetable entry is 64bit long. The first 32bit contains the lower part of the mac address, the second 32bit contains the higher part of the mac address and the valid bit (Bit 16).	R

7.4.48 RPTP_ID — PTP Core ID Register

Address: 4404 C000h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ID															
Value after reset	0	0	0	1	0	1	0	1	1	0	0	0	1	0	0	0

Table 7.49 RPTP_ID Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b16	Reserved	Set to zero on Write. ignore on Read.	R
b15 to b0	ID	ID Reading this register returns the value 0x1588 if the device is up and running	R

7.4.49 RPTP_TST — PTP Core Test Register

Address: 4404 C004h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TST															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 7.50 RPTP_TST Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b16	Reserved	Set to zero on Write. ignore on Read.	R
b15 to b0	TST	TEST 2 Write any value to this register. It will be returned XORed by the value 0x1588 on the next read.	R/W

7.4.50 RPTP_VER — PTP Version Register

Address: 4404 C008h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	IJV				IIV				FJV				FIV			
Value after reset	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	1

Table 7.51 RPTP_VER Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b16	Reserved	Set to zero on Write. ignore on Read.	R
b15 to b12	IJV	INTERFACE MAJOR VERSION	R
b11 to b8	IIV	INTERFACE MINOR VERSION	R
b7 to b4	FJV	PTP MAJOR VERSION	R
b3 to b0	FIV	PTP MINOR VERSION	R

7.4.51 RPTP_GPO — General Purpose Register

Address: 4404 C00Ch

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	OSE	PDN								—	RPT
Value after reset	X	X	X	X	X	0	0	0	0	0	0	0	0	0	X	0

Table 7.52 RPTP_GPO Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b11	Reserved	Set to zero on Write. ignore on Read.	R
b10	OSE	ONE-STEP ENABLE 0: Two-Step operation. 1: One-Step operation enabled If a frame is received while this bit changes, you may get invalid timestamps. Soft-reset recommended after change.	R/W
b9 to b2	PDN	PTPv2 DOMAIN NUMBER 255 = accept all PTP domains. Recommended for a TC, to correct the Sync residence time for all PTP domains.	R/W
b1	Reserved	Set to zero on Write. ignore on Read.	R/W
b0	RPT	RESET PTP 0: Disable 1: Soft Reset of the PTP The soft reset completion takes about 1.5μs.	R/W

7.4.52 RPTP_INT — PTP Interrupt Register

These interrupts are fired when a frame pass through each PORT and the timestamp is gotten.

Address: 4404 C014h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	B	A	I	CPU
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	0	0	0	0

Table 7.53 RPTP_INT Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b4	Reserved	Set to zero on Write. ignore on Read.	R
b3	B	PORT B INTERRUPT 1 = interrupt fired (read to clear)	R
b2	A	PORT A INTERRUPT 1 = interrupt fired (read to clear)	R
b1	I	INTERLINK INTERRUPT 1 = interrupt fired (read to clear)	R
b0	CPU	CPU INTERRUPT 1 = interrupt fired (read to clear)	R

7.4.53 RPTP_INT_MSK — PTP Interrupt Mask Register

Address: 4404 C018h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	B	A	I	CPU
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	0	0	0	0

Table 7.54 RPTP_INT_MSK Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b4	Reserved	Set to zero on Write. ignore on Read.	R
b3	B	PORT B INTERRUPT 0: Disable Interrupt 1: Enable Interrupt	R/W
b2	A	PORT A INTERRUPT 0: Disable Interrupt 1: Enable Interrupt	R/W
b1	I	INTERLINK INTERRUPT 0: Disable Interrupt 1: Enable Interrupt	R/W
b0	CPU	CPU INTERRUPT 0: Disable Interrupt 1: Enable Interrupt	R/W

7.4.54 RPTP_TS_STAT_[m] — Timestamp Status Register P[m] (m = 1 to 4)

It corresponds that m = 1: CPU, m = 2: Interlink, m = 3: PortA, m = 4: PortB

Address: 4404 C044h + 18h × (m − 1)

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	EN	—	—	TRP					DE				—	—	—	—
Value after reset	0	X	X	0	0	0	0	0	0	0	0	0	X	X	X	X

Table 7.55 RPTP_TS_STAT_[m] Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b16	Reserved	Set to zero on Write. ignore on Read.	R
b15	EN	TIMESTAMP ENABLE Enable timestamping	R/W
b14, b13	Reserved	Set to zero on Write. ignore on Read.	R
b12 to b8	TRP	TIMESTAMP READ POSITION Pointer to the read position, RPTP_TS_RD_[m], of the Timestamp	R
b7 to b4	DE	DISCARDED EVENTS Number of timestamps discarded due to insufficient memory. This is reset to 0 after reading the register.	R
b3 to b0	Reserved	Set to zero on Write. ignore on Read.	R

7.4.55 RPTP_TS_RD_[m] — Timestamp Read Register P[m] (m = 1 to 4)

It corresponds that m = 1: CPU, m = 2: Interlink, m = 3: PortA, m = 4: PortB

Address: 4404 C048h + 18h × (m − 1)

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TS															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 7.56 RPTP_TS_RD_[m] Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b16	Reserved	Set to zero on Write. ignore on Read.	R
b15 to b0	TS	TIMESTAMP Timestamp read register (11 × 16-bit read = 176 Bit) ● MSB First Bit 175 … 173 Message type 000b: No valid PTP frame 001b: PTP Sync Message 010b: PTP Pdelay_Req Message 011b: PTP Pdelay_Resp Message 100b: PTP Delay_Req Message Bit 172 …171 Timestamp receive source Port Bit 170 …166 TC Timestamp (5bit seconds) Bit 165 …136 TC Timestamp (30bit nanoseconds) Bit 135 …126 BC Timestamp (10bit seconds) Bit 125 …96 BC Timestamp (30bit nanoseconds) Bit 95 …16 The sourcePortIdentity extracted from the PTP telegram Bit 15 …0 The sequenceID extracted from the PTP telegram	R

7.4.56 RPTP_PORT_CONF_[m] — Port Config Register P[m] (m = 1 to 4)

It corresponds that m = 1: CPU, m = 2: Interlink, m = 3: PortA, m = 4: PortB

Address: 4404 C04Ch + 18h × (m − 1)

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	PDP		—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	0	0	X	X

Table 7.57 RPTP_PORT_CONF_[m] Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b4	Reserved	Set to zero on Write. ignore on Read.	R
b3, b2	PDP	PEER DELAY POSITION Pointer to the write position of RPTP_P_DELAY_[m]	R
b1, b0	Reserved	Set to zero on Write. ignore on Read.	R/W

7.4.57 RPTP_P_DELAY_[m] — Peer Delay Register P[m] (m = 1 to 4)

It corresponds that m = 1: CPU, m = 2: Interlink, m = 3: PortA, m = 4: PortB

Address: 4404 C050h + 18h × (m − 1)

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PED															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 7.58 RPTP_P_DELAY_[m] Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b16	Reserved	Set to zero on Write. ignore on Read.	R
b15 to b0	PED	PEER DELAY P2P Delay write register (4 × 16bit write = 64 Bit) (same time format as the PTP correction field) When sync frame transferred from Port n to Port m, RPTP_P_DELAY_[m] value is added to CorrectionField of the sync frame. (n=1 to 4, excluding n=m)	W

7.4.58 RPTP_PHY_DLY_TX_[m] — PHY Tx Delay Register P[m] (m = 1 to 4)

It corresponds that m = 1: CPU, m = 2: Interlink, m = 3: PortA, m = 4: PortB

Address: 4404 C054h + 18h × (m − 1)

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PDT															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 7.59 RPTP_PHY_DLY_TX_[m] Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b16	Reserved	Set to zero on Write. ignore on Read.	R
b15 to b0	PDT	PHY DELAY TX Number of ns which has to be added to the Timestamp because the time was taken too early. The value is a 16 Bit signed integer.	R/W

7.4.59 RPTP_PHY_DLY_RX_[m] — PHY Rx Delay Register P[m] (m = 1 to 4)

It corresponds that m = 1: CPU, m = 2: Interlink, m = 3: PortA, m = 4: PortB

Address: 4404 C058h + 18h × (m − 1)

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PDR															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 7.60 RPTP_PHY_DLY_RX_[m] Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b16	Reserved	Set to zero on Write. ignore on Read.	R
b15 to b0	PDR	PHY DELAY RX Number of ns which has to be subtracted from the Timestamp because the time was taken too late. The value is a 16 Bit signed integer.	R/W

7.4.60 RPTP_BUF_STAT — Buffer Status Register

These bits show 1 while a timestamp is stored in the buffer. It turned to 0 after the buffer becomes empty by reading RPTP_TS_RD register.

Address: 4404 C0A4h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	B	A	I	CPU
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	0	0	0	0

Table 7.61 RPTP_BUF_STAT Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b4	Reserved	Set to zero on Write. ignore on Read.	R
b3	B	PORT B TIMESTAMP READY A timestamp is available on this port.	R
b2	A	PORT A TIMESTAMP READY A timestamp is available on this port.	R
b1	I	INTERLINK TIMESTAMP READY A timestamp is available on this port.	R
b0	CPU	CPU TIMESTAMP READY A timestamp is available on this port.	R

7.5 Operation

7.5.1 HSR Initializing Flow

7.5.1.1 Initializing

The initialization sequence in this section is an example for preparation of system environments for using HSR under configuration below.

HSR configuration of this example:

- 2 ports are connected to external port through MII Converter.
- Interlink port is connected to GMAC2.
- GMAC1_PTP_TIMESTAMP_O signal of GMAC1 is used.

If you want to use PTP, please also refer to “8.2.2 Selecting of clocks for PTP” in the “User’s Manual: System Introduction”. The PTP timer is embedded in the GMAC1 and is configured to distribute to other modules. If PTPMCTRL. PTP_MODE[3:0] is 0, no clock is supplied for PTP operation. Reset release by PWRCTRL_SWITCHCTRL.RSTN_CLK25 is also required.

Note 1. Described in RZ/N1 User's Manual: System Introduction, Multiplexing, Electrical and Mechanical Information
When disconnect, should re-reset HSR_CLK50 domain (PWRCTRL_HSR.RSTN_C = 0).

Note 2. Should read the Alive Test 1 Register after the mode control register (MODCTRL.SW_MODE[4:0]) is set to HSR mode (the value is any of 28 to 31).

Figure 7.3 Initializing Flowchart

7.5.1.2 HSR_IP_INIT

HSR_IP_INIT sequence is used for initializing of HSR.

For HSR_IP_INIT operation, complete the following flowchart:

Note 1. HSR Switch

Figure 7.4 HSR_IP_INIT Flowchart

7.6 Usage Notes

7.6.1 PTP Registers Assignment

A part of PTP registers such as RPTP_TS_STAT_[m] are assigned to each ports, where [m] means one of the value 1 to 4. The assignments are shown below.

- m = 1: CPU
- m = 2: Interlink
- m = 3: Port A
- m = 4: Port B

7.6.2 Restriction

- HW-RTOS GMAC and HSR cannot use simultaneously.
- HSR can be used in RZ/N1D only.
- When transfer A Frame and B Frame (both have HSR TAG), both A Frame and B Frame are exactly the same (LAN ID field of HSR TAG is also the same).
  → It is not possible to have different PathID's on A- and B-Frame. In the standard is a suggestion that one could use LSB of PathID field (the laneID bit) to record on which port the frame was received, but this is optional and not implemented in this module.

Section 8 Sercos III Slave Controller

8.1 Overview

- Hardware Slave controller: Sercos III IP version 4.12.0
- 2 ports
- The serial interface operates with 100 Mbaud.
- Dual ports RAM for control and communication data.
- Telegram processing for automatic transmission, and monitoring of synchronization telegrams and data telegrams. Only transmission data which is intended for the particular interface user is processed. The transmitted data is either stored in the internal RAM. The transmission of service channel information over several communication cycles is executed automatically.
- Switch over function between Sercos protocol and standard Ethernet protocol via multiplexer.
- Monitors the received data stream to detect the frame type and starts operation when a Sercos III frame type is detected,
- Handling of the data transfers to and from SRAM based on telegram type (MST/MDT or AT) and operation mode
- Interface
  - Native mode MII
  - No Native RMII, managed by RMII/RGMII convertor (connected on external pins)

Figure 8.1 SERCOS3 Block Diagram

8.2 Signal Interface

Table 8.1 Signal Interface of the Sercos III Slave Controller (excluding PHY MII pins)

Signal Name	I/O	Description	Active
Clock			
SERCOS_HCLK	I	AHB clock	
SERCOS_CLK100	I	100 MHz clock	
SERCOS_CLK50	I	50 MHz clock	
Interrupt			
SERCOS3_DIVCLK_Int	O	Divided communication clock out, pulse sensitive	High
SERCOS3_CONCLK_Int	O	Communication synchronized control clock output, pulse sensitive	High
SERCOS3_int[0]	O	Port1 interrupt, level sensitive	High
SERCOS3_int[1]	O	Port2 interrupt, level sensitive	High
External Signal			
S3_LED_GN	O	SERCOS3 Drive output for LED (green)	High
S3_LED_RD	O	SERCOS3 Drive output for LED (red)	High
S3_ACTLEDP[1]	O	SERCOS3 Drive output for activity LED (port A) (Port 1 of SERCOS3 module)	High
S3_ACTLEDP[2]	O	SERCOS3 Drive output for activity LED (port B) (Port 2 of SERCOS3 module)	High
S3_LINKLEDP[1]	O	SERCOS3 Drive output for link LED (port A) (Port 1 of SERCOS3 module)	High
S3_LINKLEDP[2]	O	SERCOS3 Drive output for link LED (port B) (Port 2 of SERCOS3 module)	High
S3_CONCLK	O	SERCOS3 Communication synchronized control clock (CON_CLK) output	—
S3_DIVCLK	O	SERCOS3 Divided communication clock (DIV_CLK) out	—
S3_MII_LINKP[1]	I	SERCOS3 Link signal input (port A) (Port 1 of SERCOS3 module)	High*[1]
S3_MII_LINKP[2]	I	SERCOS3 Link signal input (port B) (Port 2 of SERCOS3 module)	High*[1]
S3_PHY_RESET_N	O	SERCOS3 PHY RESET	Low
S3_TESTPIN[n] (n = 1..2)	O	SERCOS3 Test signal outputs	—

Note 1. S3_MII_LINKP[2:1] active level is controlled by Ethernet PHY Link Mode register.

8.3 Register Map

Table 8.2 SERCOS III Register Map (1/2)

Address	Register Symbol	Register Name
4402 0000h	IDR	Identification Register
4402 0004h	GCSFR	Global Control / Status / Feature Register
4402 0008h	IER0	Interrupt Enable Register
4402 0010h	IMR0	Interrupt Multiplex Register
4402 0018h	IRR0	Interrupt Reset / Status Register
4402 0020h	DFCSR	Data Flow Control / Status Register
4402 0024h	PHASESR	Phase Status Register
4402 0028h	TGSR1	Telegram Reset / Status Register Port 1
4402 002Ch	TGSR2	Telegram Reset / Status Register Port 2
4402 0030h	DESCR	Descriptor Control Register
4402 0034h	STRBR	System Timer Read Back Register
4402 0038h	TCSR	Timing Control / Status Register
4402 003Ch	TRDLY	Ring Delay Register
4402 0040h	TDMST1	Time Delay MST Port 1
4402 0044h	TDMST2	Time Delay MST Port 2
4402 0048h	SCR1	Sync Time Register Port 1
4402 004Ch	SCR2	Sync Time Register Port 2
4402 0050h	SVCCSR	SVC Control / Status
4402 0054h	DTDIVCLK	Delay Time for DIV_CLK
4402 0058h	TDIV_NDIVCLK	DIV_CLK Time / Count Register
4402 005Ch	S3LED	Sercos III LED Control
4402 0060h	ASCR0	Address Segment Control Register 0 (not used)
4402 0064h	ASCR1	Address Segment Control Register 1 (not used)
4402 0068h	WDCSR	Watchdog Control & Status
4402 006Ch	WDCNT	Watchdog Counter
4402 0070h	SFCR	Sercos Frame Control
4402 0074h	MIICSR	MDIO Control / Status Register
4402 0078h	DBGOCR	Debug output control
4402 007Ch	SEQCNT	Sequence Counter
4402 0080h	MAC1P1_0	MAC Address 0
4402 0084h	MAC1P1_1	MAC Address 1
4402 0090h	IPCSR1	IP Status / Control register port 1
4402 0094h	IPCSR2	IP Status / Control register port 2
4402 0098h	IPRRS1	IP Rx RAM Segment port 1
4402 009Ch	IPRRS2	IP Rx RAM Segment port 2
4402 00A0h	IPRXS1	IP Receive Stack port 1
4402 00A4h	IPRXS2	IP Receive Stack port 2
4402 00A8h	IPTXS1	IP Transmit Stack port 1
4402 00ACh	IPTXS2	IP Transmit Stack port 2
4402 00B0h	IPLASTFL	Remaining frame length
4402 00C0h	IPFRXOK	IP aFramesReceivedOK
4402 00C4h	IPFTXOK	IP aFramesTransmittedOK
4402 00C8h	IPFCSERR	IP aFCSErrors
4402 00CCh	IPALGNERR	IP aAlignmentErrors

Table 8.2 SERCOS III Register Map (2/2)

Address	Register Symbol	Register Name
4402 00D0h	IPDISRXB	IP aDiscardResRxBuf
4402 00D4h	IPDISCLB	IP aDiscardResColBuf
4402 00D8h	IPCHVIOL	IP aIPChannelViolation
4402 00DCh	aSercosErrorCount	Sercos III Error Counter
4402 00E0h	MSTLMAX	Maximum sequential MST losses
4402 00E4h	MSTLSUM	Sum MST losses
4402 00E8h	MSTLACT	Actual sequential MST losses
4402 0100h	TMDSCL	Timing Descriptor Lower
4402 0104h	TMDSCU	Timing Descriptor Upper
4402 0108h	TMDSCSEL	Timing Descriptor Select
4402 0110h	PTMDSCL	Port Timing Descriptor Lower
4402 0114h	PTMDSCU	Port Timing Descriptor Upper
4402 0118h	PTMDSCSEL	Port Timing Descriptor Select
4402 0120h	PLLCSR	PLL Control & Status Register (not used)
4402 0124h	TCNTCYCR	TCNT Cycletime (not used)
4402 0130h	STNS	System Time (nanoseconds)
4402 0134h	STSEC	System Time (seconds)
4402 0138h	STNSTSR	System Time TSRef (nanoseconds)
4402 013Ch	STSECTSR	System Time TSRef (seconds)
4402 0140h	SCCAB	Subcycle Counter Control & Status
4402 0150h	SCCMDT	Subcycle Counter MDT
4402 0180h + 4h × n	RXBUF[n]_P1A (n = 0..2)	Receive buffer [n] base address for port 1 and buffer system A
4402 0190h + 4h × n	RXBUF[n]_P1B (n = 0..2)	Receive buffer [n] base address for port 1 and buffer system B
4402 019Ch	RXBUF_P1SVC	Receive buffer base address for service channel data at port 1
4402 01A0h + 4h × n	RXBUF[n]_P2A (n = 0..2)	Receive buffer [n] base address for port 2 and buffer system A
4402 01B0h + 4h × n	RXBUF[n]_P2B (n = 0..2)	Receive buffer [n] base address for port 2 and buffer system B
4402 01BCh	RXBUF_P2SVC	Receive buffer base address for service channel data at port 2
4402 01C0h + 4h × n	TXBUF[n]_A (n = 0..3)	Transmit buffer [n] base address for buffer system A
4402 01D0h + 4h × n	TXBUF[n]_B (n = 0..3)	Transmit buffer [n] base address for buffer system B
4402 01F0h	TXBUF_P1	Transmit buffer base address for port 1 only
4402 01F4h	TXBUF_P2	Transmit buffer base address for port 2 only
4402 01FCh	TXBUF_SVC	Transmit buffer base address for service channel data
4402 0200h	RXBUFCSR_A	Receive buffer control buffer system A
4402 0204h	RXBUFTV_A	Rx Buffer Telegram Valid A
4402 0208h	RXBUFTR_A	Rx Buffer Telegram Requirements A
4402 020Ch	TXBUFCSR_A	Transmit buffer control for buffer system A
4402 0210h	RXBUFCSR_B	Receive buffer control buffer system B
4402 0214h	RXBUFTV_B	Rx Buffer Telegram Valid B
4402 0218h	RXBUFTR_B	Rx Buffer Telegram Requirements B
4402 021Ch	TXBUFCSR_B	Transmit buffer control for buffer system B

8.4 Register Description

8.4.1 IDR — Identification Register

Address: 4402 0000h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	S3FrameType															
Value after reset	1	0	0	0	1	0	0	0	1	1	0	0	1	1	0	1

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DeviceType			Testversion			Version					Release				
Value after reset	0	0	1	0	0	0	0	0	1	0	0	0	1	1	0	0

Table 8.3 IDR Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b16	S3FrameType	Sercos III Ethernet Type (0x88CD)	R
b15 to b13	DeviceType	0: Master device 1: Slave device	R
b12 to b10	Testversion	Testversion. Production versions must have a 0 here. A value greater than 0 is for testing only.	R
b9 to b5	Version	Device version	R
b4 to b0	Release	Device release	R

8.4.2 GCSFR — Global Control / Status / Feature Register

Address: 4402 0004h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	HardwareVersionBus								—	—	—	—	Size_of_memory_banks			
Value after reset	0	0	0	0	0	1	0	0	X	X	X	X	0	1	1	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	Line_break_mode	—	—	—	Line_break_sensitivity				—	—	—	DescriptorFeedback	BCastDis	BCastRed	PHYReset	SR
Value after reset	0	X	X	X	0	1	0	1	X	X	X	0	0	0	1	0

Table 8.4 GCSFR Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b24	HardwareVersionBus	Specifies the hardware version of the bus interface.	R
b23 to b20	Reserved	Reserved	R
b19 to b16	Size_of_memory_banks	Specifies the size of the memory bank when address space is segmented. size is $2^{(8+n)}$ (byte)	R
b15	Line_break_mode	Line break detection delayed to following MST loss event.	R/W
b14 to b12	Reserved	Reserved	R
b11 to b8	Line_break_sensitivity	Linebreak sensitivity (RxErr and MII false carrier). Defines how fast a linebreak is detected. Increment value for 8 bit linebreak up/down counter. Count frequency is 12.5 MHz; linebreak is detected at overflow of counter.	R/W
b7 to b5	Reserved	Reserved	R
b4	DescriptorFeedback	Enable usage of descriptor enable feedback status bits inside DFCSR (activation for compatibility reasons).	R/W
b3	BCastDis	Disable forwarding of Sercos III frames during NRT and to inactive port of loopback slave (Isolation mode).	R/W
b2	BCastRed	Reduce amount of forwarded Sercos III frames to inactive port of a loopback slave. Port timer event for opening MST header window is required.	R/W
b1	PHYReset	PHY Reset. Writing a 1 initiates a reset of the connected S3_PHY_RESET_N. The duration of the reset is between 65 ms and 130 ms. This bit is cleared automatically after an additional period of 200 ms to ensure accessibility to the PHY registers.	R/W
b0	SR	Software Reset. Writing a 1 initiate software reset on the SERCOS3 module. The duration of the reset is around 2 microseconds. Reading this bit reflects the reset status of the device. When read as 1 reset is still active. This bit is also set when the external reset is active.	R/W

8.4.3 IER0 — Interrupt Enable Register

Address: 4402 0008h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	IE															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	IE															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 8.5 IER0 Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	IE	Enables Interrupt Source	R/W

8.4.4 IMR0 — Interrupt Multiplex Register

Address: 4402 0010h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	IM															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	IM															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 8.6 IMR0 Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	IM	Assigns interrupt source to interrupt output	R/W

8.4.5 IRR0 — Interrupt Reset/Status Register

Address: 4402 0018h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	IR															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	IR															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 8.7 IRR0 Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	IR	(Read) Status of interrupt source (Write) Reset interrupt source	R/W

8.4.6 DFCSR — Data Flow Control/Status Register

Address: 4402 0020h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	RxEnableFeedback	TxEnableFeedback	—	—	P2_Rx_all_done	P2_Tx_all_done	P1_Rx_all_done	P1_Tx_all_done	Swap_Counters	Line_Topology	NRT_Forward	Ring_Topology	Port2_Link	Port1_Link	Port2_Line_Status	Port1_Line_Status
Value after reset	0	0	X	X	0	1	0	1	0	0	1	0	0	0	1	1

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	Topology_address_increment				Tx_MDT_Enable	Disable Length Ctrl	Rx_Enable	Tx_Enable	Topology_Write_Mode	Automatic_topology_detection	Topology_Primary_Secondary		State_of_Inactive_Port		Topology_Port	
Value after reset	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	1

Table 8.8 DFCSR Register Contents (1/2)

Bit Position	Bit Name	Function	R/W
b31	RxEnableFeedback	Feedback of internal enable of the Rx Descriptor Units 0: Descriptor Unit disabled 1: Descriptor Unit enabled (do not change descriptor list and buffer offsets)	R
b30	TxEnableFeedback	Feedback of internal enable of the Tx Descriptor Units 0: Descriptor Unit disabled 1: Descriptor Unit enabled (do not change descriptor list and buffer offsets)	R
b29, b28	Reserved	Reserved	R
b27	P2_Rx_all_done	Rx buffer change has finished for all systems	R
b26	P2_Tx_all_done	Tx buffer change has finished for all systems	R
b25	P1_Rx_all_done	Rx buffer change has finished for all systems	R
b24	P1_Tx_all_done	Tx buffer change has finished for all systems	R
b23	Swap_Counters	0: P/S interpretation equal to Port 1/Port 2 1: P/S interpretation inverse to Port 1/Port 2	R
b22	Line_Topology	0: Ring topology or no MST's 1: Receiving P- or S-MST's at both ports.	R
b21	NRT_Forward	0: Topology is 3 and ring topology and CP>0 (info for disabling ColBuf) 1: Info for enabling Collision Buffer.	R
b20	Ring_Topology	0: Line topology or no MST's 1: Receiving P- and S-MST's at different ports.	R
b19	Port2_Link	Link attached	R
b18	Port1_Link	Link attached	R
b17	Port2_Line_Status	0: No error on line 1: Error detected	R
b16	Port1_Line_Status	0: No error on line 1: Error detected	R
b15 to b12	Topology_address_increment	This value is added to the topology address field. Increment the topology address by a value of 1..15 at Fast-Forward and inactive port. At the Loopback Port, the increment is twice minus 1 (1..29) (multi slaves).	R/W
b11	Tx_MDT_Enable	Produce enable for MDT's. If this bit is set, there are 4 new descriptor list pointers after AT list pointers.	R/W
b10	DisableLengthCtrl	Disable packet length check	R/W
b9	Rx_Enable	Enable the Rx Descriptor Units	R/W
b8	Tx_Enable	Enable the Tx Descriptor Units	R/W

Table 8.8 DFCSR Register Contents (2/2)

Bit Position	Bit Name	Function	R/W
b7	Topology_Write_Mode	Topology Update Mode 0: Set new topology in port mode 1: Set new topology in P/S mode	R/W
b6	Automatic_topology_detection	Enable automatic topology detection. Similar mechanism to CP0. Port deactivation after single MST loss.	R/W
b5, b4	Topology_Primary_Secondary	Set topology (Write) Get current topology (Read) 0: Fast-Forward on both ports 1: Loopback with Forward of P-Telegrams 2: Loopback with Forward of S-Telegrams 3: NRT mode (store and forward)	R/W
b3, b2	State_of_Inactive_Port	Device Status 0: No Link on inactive port 1: Link at inactive port 2: P-Telegram on inactive port 3: S-Telegram on inactive port	R/W
b1, b0	Topology_Port	Set topology (Write) Get current topology (Read) 0: Forward Port 1 to Port 2 and Port 2 to Port 1. Change on Error. (RT Mode) 1: Loopback on Port 2 and forward to Port 1. No change on Error (RT Mode) 2: Loopback on Port 1 and forward to Port 2. No change on Error (RT Mode) 3: Forward Port 1 to Port 2 and Port 2 to Port 1. No change on Error. (NRT Mode)	R/W

8.4.7 PHASESR — Phase Status Register

Address: 4402 0024h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	Valid	—	—	—	—	—	—	—	CPS	—	—	—	MST_Phase			
Value after reset	0	X	X	X	X	X	X	X	0	X	X	X	0	0	0	0

Table 8.9 PHASESR Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b16	Reserved	Reserved	R
b15	Valid	Bit 0-3 and Bit 7 are valid (reset to 0 after write)	R/W
b14 to b8	Reserved	Reserved	R
b7	CPS	Current Phase switch bit set in MST	R
b6 to b4	Reserved	Reserved	R
b3 to b0	MST_Phase	Current Phase information out of MST	R

8.4.8 TGSR1 — Telegram Reset/Status Register Port 1

Address: 4402 0028h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	First_MST	CycCnt_valid	Cycle_count			—	—	—	—	—	—	—	—
Value after reset	X	X	X	0	0	0	0	0	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	AT0_misss	MST_double_miss	MST_miss	MST_window_error	Primary_Secondary	MST_valide	AT3	AT2	AT1	AT0	MDT3	MDT2	MDT1	MDT0
Value after reset	X	X	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 8.10 TGSR1 Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b29	Reserved	Reserved	R
b28	First_MST	Indicates that the first MST arrives at this port	R
b27	CycCnt_valid	Cycle count valid bit from the MST header of MDT0	R
b26 to b24	Cycle_count	Cycle count from the MST header of MDT0	R
b23 to b14	Reserved	Reserved	R
b13	AT0_misss	AT0 was missed inside AT0 window or CRC invalid	R/W
b12	MST_double_miss	MST was missed for 2 times consecutively	R/W
b11	MST_miss	MST missed or CRC invalid	R/W
b10	MST_window_error	MST received, CRC valid, out of MST receive window	R/W
b9	Primary_Secondary	Primary / Secondary telegram	R
b8	MST_valide	MST received, CRC valid	R/W
b7	AT3	AT 3 received, FCS valid	R/W
b6	AT2	AT 2 received, FCS valid	R/W
b5	AT1	AT 1 received, FCS valid	R/W
b4	AT0	AT 0 received, FCS valid	R/W
b3	MDT3	MDT 3 received, FCS valid	R/W
b2	MDT2	MDT 2 received, FCS valid	R/W
b1	MDT1	MDT 1 received, FCS valid	R/W
b0	MDT0	MDT 0 received, FCS valid	R/W

8.4.9 TGSR2 — Telegram Reset/Status Register Port 2

Address: 4402 002Ch

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	First_MST	CycCnt_valid	Cycle_count			—	—	—	—	—	—	—	—
Value after reset	X	X	X	0	0	0	0	0	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	AT0_misss	MST_double_miss	MST_miss	MST_window_error	Primary_Secondary	MST_valide	AT3	AT2	AT1	AT0	MDT3	MDT2	MDT1	MDT0
Value after reset	X	X	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 8.11 TGSR2 Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b29	Reserved	Reserved	R
b28	First_MST	Indicates that the first MST arrives at this port	R
b27	CycCnt_valid	Cycle count valid bit from the MST header of MDT0	R
b26 to b24	Cycle_count	Cycle count from the MST header of MDT0	R
b23 to b14	Reserved	Reserved	R
b13	AT0_misss	AT0 was missed inside AT0 window or CRC invalid	R/W
b12	MST_double_miss	MST was missed for 2 times consecutively	R/W
b11	MST_miss	MST missed or CRC invalid	R/W
b10	MST_window_error	MST received, CRC valid, out of MST receive window	R/W
b9	Primary_Secondary	Primary / Secondary telegram	R
b8	MST_valide	MST received, CRC valid	R/W
b7	AT3	AT 3 received, FCS valid	R/W
b6	AT2	AT 2 received, FCS valid	R/W
b5	AT1	AT 1 received, FCS valid	R/W
b4	AT0	AT 0 received, FCS valid	R/W
b3	MDT3	MDT 3 received, FCS valid	R/W
b2	MDT2	MDT 2 received, FCS valid	R/W
b1	MDT1	MDT 1 received, FCS valid	R/W
b0	MDT0	MDT 0 received, FCS valid	R/W

8.4.10 DESCR — Descriptor Control Register

Address: 4402 0030h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	Tx_IDXTBL_Offset											—	—
Value after reset	X	X	X	0	0	0	0	0	0	0	0	0	0	0	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	Rx_IDXTBL_Offset											—	—
Value after reset	X	X	X	0	0	0	0	0	0	0	0	0	0	0	X	X

Table 8.12 DESCR Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b29	Reserved	Reserved	R
b28 to b18	Tx_IDXTBL_Offset	Points to an offset list inside Tx RAM. Each entry of this list is assigned to a different Sercos III frame from AT0 to AT3.	R/W
b17 to b13	Reserved	Reserved	R
b12 to b2	Rx_IDXTBL_Offset	Points to an offset list inside Rx RAM. Each entry of this list is assigned to a different Sercos III frame from MDT0 to AT3.	R/W
b1, b0	Reserved	Reserved	R

8.4.11 STRBR — System Timer Read Back Register

Address: 4402 0034h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	TCNT									
Value after reset	X	X	X	X	X	X	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TCNT															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 8.13 STRBR Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b26	Reserved	Reserved	R
b25 to b0	TCNT	TCNT counter value	R/W

8.4.12 TCSR — Timing Control/Status Register

Address: 4402 0038h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	SysTimeUpdate	—	—	—	—	—	—	—	—	—	—	—	—	—	P2_MST_Dis	P1_MST_Dis
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TimeSync_Enable	DivOD	DivClkPol	DivClk_mode	—	CONOE	CON_Pol	CON_En	—	—	—	—	ET3	—	ET1_2	ET0
Value after reset	0	0	0	0	X	0	0	0	0	0	0	0	0	0	0	0

Table 8.14 TCSR Register Contents

Bit Position	Bit Name	Function	R/W
b31	SysTimeUpdate	System time update information. 0: System time was not updated since last acknowledgement. 1: System time was updated since last acknowledgement. Acknowledge with writing 1 to this bit.	R/W
b30 to b18	Reserved	Reserved	R
b17	P2_MST_Dis	Port2 MST disable 0: MST Signal of Port2 is used for synchronization 1: MST Signal of Port2 is not used for synchronization	R/W
b16	P1_MST_Dis	Port1 MST disable 0: MST Signal of Port1 is used for synchronization 1: MST Signal of Port1 is not used for synchronization	R/W
b15	TimeSync_Enable	Enables synchronization of system time	R/W
b14	DivOD	DIV_CLK output disable (not used)	R/W
b13	DivClkPol	DIVCLK polarity 0: DIV_CLK is active on positive edge 1: DIV_CLK is active on negative edge	R/W
b12	DivClk_mode	DIVCLK modes 0: DIV_CLK mode 0 1: DIV_CLK mode 1	R/W
b11	Reserved	Reserved	R
b10	CONOE	Output Enable of output CON_CLK (not used)	R/W
b9	CON_Pol	Polarity of output CON_CLK 0: active high 1: active low	R/W
b8	CON_En	Enable of CON_CLK	R/W
b7 to b4	Reserved	Reserved	R
b3	ET3	Enable system time increment	R/W
b2	Reserved	Reserved	R
b1	ET1_2	Enable timer for port 1/2	R/W
b0	ET0	Enable main timer TCNT	R/W

8.4.13 TRDLY — Ring Delay Register

Address: 4402 003Ch

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	Ringdelay			
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	Ringdelay															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 8.15 TRDLY Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b20	Reserved	Reserved	R
b19 to b0	Ringdelay	Ring delay of Sercos III	R/W

8.4.14 TDMST1 — Time Delay MST Port 1

Address: 4402 0040h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	Time_Delay_MST			
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	Time_Delay_MST															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 8.16 TDMST1 Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b20	Reserved	Reserved	R
b19 to b0	Time_Delay_MST	Time Delay MST Port 1	R/W

8.4.15 TDMST2 — Time Delay MST Port 2

Address: 4402 0044h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	Time_Delay_MST			
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	Time_Delay_MST															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 8.17 TDMST2 Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b20	Reserved	Reserved	R
b19 to b0	Time_Delay_MST	Time Delay MST Port 2	R/W

8.4.16 SCR1 — Sync Time Register Port 1

Address: 4402 0048h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	Sync_Count			
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	Sync_Count															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 8.18 SCR1 Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b20	Reserved	Reserved	R
b19 to b0	Sync_Count	Sync Time Register Port 1	R

8.4.17 SCR2 — Sync Time Register Port 2

Address: 4402 004Ch

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	Sync_Count			
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	Sync_Count															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 8.19 SCR2 Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b20	Reserved	Reserved	R
b19 to b0	Sync_Count	Sync Time Register Port 2	R

8.4.18 SVCCSR — SVC Control / Status

Address: 4402 0050h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	SVC_Busy	Process_error	Process_start	—	—	—	MDT_select		Port_select	SVC_enable
Value after reset	X	X	X	X	X	X	0	0	0	X	X	X	0	0	0	0

Table 8.20 SVCCSR Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b10	Reserved	Reserved	R
b9	SVC_Busy	The SVC machine is busy	R/W
b8	Process_error	Process error occurred while MDT processing (read) clear process error (write)	R/W
b7	Process_start	Start SVC machine manually with positive edge of bit location (for debug use)	R/W
b6 to b4	Reserved	Reserved	R
b3, b2	MDT_select	Last MDT that contains relevant SVC data	R/W
b1	Port_select	Trigger SVC machine from port1 (0) or port 2 (1)	R/W
b0	SVC_enable	Enable service-channel operation	R/W

8.4.19 DTDIVCLK —Delay Time for DIV_CLK

Address: 4402 0054h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	DTDIVCLK									
Value after reset	X	X	X	X	X	X	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DTDIVCLK															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 8.21 DTDIVCLK Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b26	Reserved	Reserved	R
b25 to b0	DTDIVCLK	Delay Time for DIV_CLK (unit: ns)	R/W

8.4.20 TDIV_NDIVCLK — DIV_CLK Time / Count Register

Address: 4402 0058h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	NDIVCLK								TDIVCLK							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TDIVCLK															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 8.22 TDIV_NDIVCLK Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b24	NDIVCLK	DIV_CLK Count	R/W
b23 to b0	TDIVCLK	DIV_CLK Time (unit: ns)	R/W

8.4.21 S3LED — Sercos III LED Control

Address: 4402 005Ch

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	Color_6		—	—	Color_5		—	CycSkip		flash_2 Hz	—	CycSplit		
Value after reset	X	X	0	0	X	X	0	0	X	0	0	0	X	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	Color_4		—	—	Color_3		—	—	Color_2		—	—	Color_1	
Value after reset	X	X	0	0	X	X	0	0	X	X	0	0	X	X	0	0

Table 8.23 S3LED Register Contents

Bit Position	Bit Name	Function	R/W
b31, b30	Reserved	Reserved	R
b29, b28	Color_6	sixth Color for watchdog, write protected if watchdog is active (dark: 0, green: 1, red: 2, orange: 3)	R/W
b27, b26	Reserved	Reserved	R
b25, b24	Color_5	fifth Color for watchdog, write protected if watchdog is active (dark: 0, green: 1, red: 2, orange: 3)	R/W
b23	Reserved	Reserved	R
b22, b21	CycSkip	Half amount of cycles to be skipped	R/W
b20	flash_2Hz	LED flashes with roughly 2 Hz instead of 4 Hz	R/W
b19	Reserved	Reserved	R
b18 to b16	CycSplit	Amount of cycles for Color_3 and Color_4 activity	R/W
b15, b14	Reserved	Reserved	R
b13, b12	Color_4	Fourth Color for second period (dark: 0, green: 1, red: 2, orange: 3)	R/W
b11, b10	Reserved	Reserved	R
b9, b8	Color_3	Third Color for second period (dark: 0, green: 1, red: 2, orange: 3)	R/W
b7, b6	Reserved	Reserved	R
b5, b4	Color_2	Second Color (dark: 0, green: 1, red: 2, orange: 3)	R/W
b3, b2	Reserved	Reserved	R
b1, b0	Color_1	First Color (dark: 0, green: 1, red: 2, orange: 3)	R/W

8.4.22 WDCSR — Watchdog Control & Status

Address: 4402 0068h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	alarm	active
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	magic_pattern															
Value after reset	1	0	0	0	1	0	0	0	1	1	0	0	1	1	0	1

Table 8.24 WDCSR Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b18	Reserved	Reserved	R
b17	alarm	watchdog alarm when actual count is zero	R/W
b16	active	watchdog is active (default is inactive after reset)	R/W
b15 to b0	magic_pattern	magic pattern to trigger watchdog (writing the magic pattern triggers the watchdog) (writing inverse magic pattern disables watchdog)	R/W

8.4.23 WDCNT — Watchdog Counter

Address: 4402 006Ch

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	actual_count															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	reset_count															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 8.25 WDCNT Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b16	actual_count	actual watchdog counter value	R
b15 to b0	reset_count	counter reset value when watchdog is triggered	R/W

8.4.24 SFCR — Sercos Frame Control

Address: 4402 0070h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	Frame_type2	PortSelect2	—	—	FrameNumber2		—	—	FrameType1	PortSelect1	—	—	FrameNumber1	
Value after reset	X	X	0	0	X	X	0	0	X	X	0	0	X	X	0	0

Table 8.26 SFCR Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b14	Reserved	Reserved	R
b13	Frame_type2	The frame type (MDT,AT) of interrupt 6	R/W
b12	PortSelect2	The reception port of interrupt 6	R/W
b11, b10	Reserved	Reserved	R
b9, b8	FrameNumber2	The Sercos III frame number of interrupt 6	R/W
b7, b6	Reserved	Reserved	R
b5	FrameType1	The frame type (MDT, AT) of interrupt 5	R/W
b4	PortSelect1	The reception port of interrupt 5	R/W
b3, b2	Reserved	Reserved	R
b1, b0	FrameNumber1	The Sercos III frame number of interrupt 5	R/W

8.4.25 MIICSR — MDIO Control / Status Register

Address: 4402 0074h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	MDIO_En_P2	MDIO_P2	MDC_P2	—	—	—	—	—	MDIO_En_P1	MDIO_P1	MDC_P1
Value after reset	X	X	X	X	X	0	1	0	X	X	X	X	X	0	1	0

Table 8.27 MIICSR Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b11	Reserved	Reserved	R
b10	MDIO_En_P2	This bit selects the MDIO direction. 0 selects input (read MDIO), 1 selects output (write MDIO).	R/W
b9	MDIO_P2	This bit has two functions. When MDIO_En_P2 is low, it reflects the state of the MDIO Pin. When MDIO_En_P2 is high, it reflects the state of the output level.	R/W
b8	MDC_P2	This bit is direct output to the PHY MII MDC Pin	R/W
b7 to b3	Reserved	Reserved	R
b2	MDIO_En_P1	This bit selects the MDIO direction. 0 selects input (read MDIO), 1 selects output (write MDIO).	R/W
b1	MDIO_P1	This bit has two functions. When MDIO_En_P1 is low, it reflects the state of the MDIO Pin. When MDIO_En_P1 is high, it reflects the state of the output level.	R/W
b0	MDC_P1	This bit is direct output to the PHY MII MDC Pin	R/W

8.4.26 DBGOCR — Debug Output Control

Address: 4402 0078h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	TS2					—	—	—	TS1				
Value after reset	X	X	X	0	0	0	0	0	X	X	X	0	0	0	0	0

Table 8.28 DBGOCR Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b13	Reserved	Reserved	R
b12 to b8	TS2	Selects the signal to be output on test pin S3_TESTPIN2	R/W
b7 to b5	Reserved	Reserved	R
b4 to b0	TS1	Selects the signal to be output on test pin S3_TESTPIN1	R/W

8.4.27 SEQCNT — Sequence Counter

Address: 4402 007Ch

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	SEQCNT_P2															
Value after reset	1	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SEQCNT_P1															
Value after reset	1	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1

Table 8.29 SEQCNT Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b16	SEQCNT_P2	Value of sequence counter field in MST's at port 2	R/W
b15 to b0	SEQCNT_P1	Value of sequence counter field in MST's at port 1	R/W

8.4.28 MAC1P1_0 — MAC Address 0

Address: 4402 0080h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	B2								B3							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	B4								B5							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 8.30 MAC1P1_0 Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b24	B2	MAC Address	R/W
b23 to b16	B3	MAC Address	R/W
b15 to b8	B4	MAC Address	R/W
b7 to b0	B5	MAC Address	R/W

8.4.29 MAC1P1_1 — MAC Address 1

Address: 4402 0084h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	B0								B1							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 8.31 MAC1P1_1 Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b16	Reserved	Reserved	R
b15 to b8	B0	MAC Address	R/W
b7 to b0	B1	MAC Address	R/W

8.4.30 IPCSR1 — IP Status/Control register port 1

Address: 4402 0090h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	Link	—	—	—	—	—	—	—	—	—	—	—	IPTxBufEmpty	IPTxBufRdy	IPRxBufFull	IPRxRdy
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	IPTxBufEmptyIntEn	IPTxBufRdyIntEn	IPRxBufFullIntEn	IPRxRdyIntEn	—	Promiscuous	ColBufDisable	MulticastDisable	BroadcastDisable	S3FrameFilter	IPRxEn	IPTxEn
Value after reset	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

Table 8.32 IPCSR1 Register Contents

Bit Position	Bit Name	Function	R/W
b31	Link	Link exists at port 1	R
b30 to b20	Reserved	Reserved	R
b19	IPTxBufEmpty	Set when IP transmit buffer is empty for port 1.	R
b18	IPTxBufRdy	Set after each transmitted frame IP transmit buffer able to accept a frame by the host.	R
b17	IPRxBufFull	Set when IP Rx-buffer of port 1 is full.	R
b16	IPRxRdy	Set when an IP Ethernet frame is received at port 1 without error.	R
b15 to b13	Reserved	Reserved	R
b12	Reserved	Should be 0	R/W
b11	IPTxBufEmptyIntEn	Set to ONE enables Interrupt Int_IPIntPort1 on event IPTxBufEmpty.	R/W
b10	IPTxBufRdyIntEn	Set to ONE enables Interrupt Int_IPIntPort1 on event IPTxBufRdy.	R/W
b9	IPRxBufFullIntEn	Set to ONE enables Interrupt Int_IPIntPort1 on event IPRxBufFull.	R/W
b8	IPRxRdyIntEn	Set to ONE enables Interrupt Int_IPIntPort1 on event IPRxRdy.	R/W
b7	Reserved	Reserved	R
b6	Promiscuous	Receive all Frames without checking the destination address.	R/W
b5	ColBufDisable	Disables collision buffer. Frames are not forwarded to opposite port.	R/W
b4	MulticastDisable	Disables reception of multicast frames in IP channel. Forwarding is not affected.	R/W
b3	BroadcastDisable	Disables reception of broadcast frames in IP channel. Forwarding is not affected.	R/W
b2	S3FrameFilter	Filter Sercos III frame of communication phase greater than zero. Those frame will not be forwarded to opposite port during NRT state.	R/W
b1	IPRxEn	Receive enable port 1, disabling the Rx port resets the internal state machine and sets the Rx-buffer pointers from the register values (IPRRS1). Therefore, IPRRS1 has to be set before IPRxEn is activated. Forwarding over collision buffer is not affected.	R/W
b0	IPTxEn	Transmit enable port 1. Disabling the Tx port resets the internal state machine and disables sending of internal IP frames. Forwarding over collision buffer is not affected. If this bit is reset during sending of an own frame, the frame will be broken.	R/W

8.4.31 IPCSR2 — IP Status/Control register port 2

Address: 4402 0094h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	Link	—	—	—	—	—	—	—	—	—	—	—	IPTxBufEmpty	IPTxBufRdy	IPRxBufFull	IPRxRdy
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	IPTxBufEmptyIntEn	IPTxBufRdyIntEn	IPRxBufFullIntEn	IPRxRdyIntEn	—	Promiscuous	ColBufDisable	MulticastDisable	BroadcastDisable	S3FrameFilter	IPRxEn	IPTxEn
Value after reset	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

Table 8.33 IPCSR2 Register Contents

Bit Position	Bit Name	Function	R/W
b31	Link	Link exists at port 2.	R
b30 to b20	Reserved	Reserved	R
b19	IPTxBufEmpty	Set when IP transmit buffer is empty for port 2.	R
b18	IPTxBufRdy	Set after each transmitted frame IP transmit buffer able to accept a frame by the host.	R
b17	IPRxBufFull	Set when IP Rx-buffer of port 2 is full.	R
b16	IPRxRdy	Set when an IP Ethernet frame is received at port 2 without error.	R
b15 to b13	Reserved	Reserved	R
b12	Reserved	Should be 0	R/W
b11	IPTxBufEmptyIntEn	Set to ONE enables Interrupt Int_IPIntPort2 on event IPTxBufEmpty.	R/W
b10	IPTxBufRdyIntEn	Set to ONE enables Interrupt Int_IPIntPort2 on event IPTxBufRdy.	R/W
b9	IPRxBufFullIntEn	Set to ONE enables Interrupt Int_IPIntPort2 on event IPRxBufFull.	R/W
b8	IPRxRdyIntEn	Set to ONE enables Interrupt Int_IPIntPort2 on event IPRxRdy.	R/W
b7	Reserved	Reserved	R
b6	Promiscuous	Receive all Frames without checking the destination address.	R/W
b5	ColBufDisable	Disables collision buffer. Frames are not forwarded to opposite port.	R/W
b4	MulticastDisable	Disables reception of multicast frames in IP channel. Forwarding is not affected.	R/W
b3	BroadcastDisable	Disables reception of broadcast frames in IP channel. Forwarding is not affected.	R/W
b2	S3FrameFilter	Filter Sercos III frame of communication phase greater than zero. Those frame will not be forwarded to opposite port during NRT state.	R/W
b1	IPRxEn	Receive enable port 2, disabling the Rx port resets the internal state machine and sets the Rx-buffer pointers from the register values (IPRRS2). Therefore, IPRRS2 has to be set before IPRxEn is activated. Forwarding over collision buffer is not affected.	R/W
b0	IPTxEn	Transmit enable port 2. Disabling the Tx port resets the internal state machine and disables sending of internal IP frames. Forwarding over collision buffer is not affected. If this bit is reset during sending of an own frame, the frame will be broken.	R/W

8.4.32 IPRRS1 — IP Rx Ram Segment Port 1

Address: 4402 0098h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	Ram_segment_Last															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	Ram_segment_First															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 8.34 IPRRS1 Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b16	Ram_segment_Last	Last segment in the Rx Ram for storing IP data	R/W
b15 to b0	Ram_segment_First	First segment in the Rx Ram for storing IP data	R/W

8.4.33 IPRRS2 — IP Rx Ram Segment Port 2

Address: 4402 009Ch

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	Ram_segment_Last															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	Ram_segment_First															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 8.35 IPRRS2 Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b16	Ram_segment_Last	Last segment in the Rx Ram for storing IP data	R/W
b15 to b0	Ram_segment_First	First segment in the Rx Ram for storing IP data	R/W

8.4.34 IPRXS1 — IP Receive Stack Port 1

Address: 4402 00A0h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	Status					Data_length										
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	Ram_segment_address															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 8.36 IPRXS1 Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b27	Status	Reserved for future use	R/W
b26 to b16	Data_length	Defines the number of received data bytes	R/W
b15 to b0	Ram_segment_address	Defines the segment position of received data in the Rx Ram	R/W

8.4.35 IPRXS2 — IP Receive Stack Port 2

Address: 4402 00A4h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	Status					Data_length										
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	Ram_segment_address															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 8.37 IPRXS2 Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b27	Status	Reserved for future use	R/W
b26 to b16	Data_length	Defines the number of received data bytes	R/W
b15 to b0	Ram_segment_address	Defines the segment position of received data in the Rx Ram	R/W

8.4.36 IPTXS1 — IP Transmit Stack Port 1

Address: 4402 00A8h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	Count			—	Data_length										
Value after reset	X	0	0	0	X	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	Ram_segment_address															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 8.38 IPTXS1 Register Contents

Bit Position	Bit Name	Function	R/W
b31	Reserved	Reserved	R
b30 to b28	Count	Amount of frames stored on stack	R/W
b27	Reserved	Reserved	R
b26 to b16	Data_length	Defines the number of data bytes to be transmitted	R/W
b15 to b0	Ram_segment_address	Defines the segment position of Tx data in the Tx Ram	R/W

8.4.37 IPTXS2 — IP Transmit Stack Port 2

Address: 4402 00ACh

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	Count			—	Data_length										
Value after reset	X	0	0	0	X	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	Ram_segment_address															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 8.39 IPTXS2 Register Contents

Bit Position	Bit Name	Function	R/W
b31	Reserved	Reserved	R
b30 to b28	Count	Amount of frames stored on stack	R/W
b27	Reserved	Reserved	R
b26 to b16	Data_length	Defines the number of data bytes to be transmitted	R/W
b15 to b0	Ram_segment_address	Defines the segment position of Tx data in the Tx Ram	R/W

8.4.38 IPLASTFL — Remaining Frame Length

Address: 4402 00B0h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	length										
Value after reset	X	X	X	X	X	0	0	0	0	0	0	0	0	0	0	0

Table 8.40 IPLASTFL Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b11	Reserved	Reserved	R
b10 to b0	length	Remaining frame length after last transmit event	R/W

8.4.39 IPFRXOK — IP aFramesReceivedOK

Address: 4402 00C0h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	aFramesReceivedOK_2															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	aFramesReceivedOK_1															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 8.41 IPFRXOK Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b16	aFramesReceivedOK_2	Counts all received frames without error on port 2 (includes forwarded and discarded frames due low resource).	R
b15 to b0	aFramesReceivedOK_1	Counts all received frames without error on port 1 (includes forwarded and discarded frames due low resource).	R

8.4.40 IPFTXOK — IP aFramesTransmittedOK

Address: 4402 00C4h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	aFramesTransmittedOK_2															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	aFramesTransmittedOK_1															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 8.42 IPFTXOK Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b16	aFramesTransmitted OK_2	Counts all transmitted frames on port 2 (includes forwarded frames).	R
b15 to b0	aFramesTransmitted OK_1	Counts all transmitted frames on port 1 (includes forwarded frames).	R

8.4.41 IPFCSERR — IP aFCSErrors

Address: 4402 00C8h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	aFCSErrors_Port_2															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	aFCSErrors_Port_1															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 8.43 IPFCSERR Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b16	aFCSErrors_Port_2	Counts the received Ethernet frames on port 2 with defective frame check sequence FCS or RxER indications.	R
b15 to b0	aFCSErrors_Port_1	Counts the received Ethernet frames on port 1 with defective frame check sequence FCS or RxER indications.	R

8.4.42 IPALGNERR — IP aAlignmentErrors

Address: 4402 00CCh

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	aFrameErrors_Port_2															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	aFrameErrors_Port_1															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 8.44 IPALGNERR Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b16	aFrameErrors_Port_2	The counter increments when a defective ethernet frame is detected on port 2. Defective frames are wrong aligned frames.	R
b15 to b0	aFrameErrors_Port_1	The counter increments when a defective ethernet frame is detected on port 1. Defective frames are wrong aligned frames.	R

8.4.43 IPDISRXB — IP aDiscardResRxBuf

Address: 4402 00D0h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	aDiscardResRxBuf_Port_2															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	aDiscardResRxBuf_Port_1															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 8.45 IPDISRXB Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b16	aDiscardResRxBuf_Port_2	The counter counts the discarded receive Ethernet frames in case of missing Rx-buffer resources on port 2.	R
b15 to b0	aDiscardResRxBuf_Port_1	The counter counts the discarded receive Ethernet frames in case of missing Rx-buffer resources on port 1.	R

8.4.44 IPDISCLB — IP aDiscardResColBuf

Address: 4402 00D4h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	aDiscardResColBuf_Port_2															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	aDiscardResColBuf_Port_1															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 8.46 IPDISCLB Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b16	aDiscardResColBuf_Port_2	The counter counts the discarded forwarding Ethernet frames in case of missing collision buffer resources on port 2.	R
b15 to b0	aDiscardResColBuf_Port_1	The counter counts the discarded forwarding Ethernet frames in case of missing collision buffer resources on port 1.	R

8.4.45 IPCHVIOL — IP aIPChannelViolation

Address: 4402 00D8h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	aIPChannelViolation_Port_2															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	aIPChannelViolation_Port_1															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 8.47 IPCHVIOL Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b16	aIPChannelViolation_Port_2	The counter increments on Ethernet frames that violate IP channel time boundaries on port 2.	R
b15 to b0	aIPChannelViolation_Port_1	The counter increments on Ethernet frames that violate IP channel time boundaries on port 1.	R

8.4.46 aSercosErrorCount — Sercos III Error Counter

Address: 4402 00DCh

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	aSercosErrorCount_Port_2															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	aSercosErrorCount_Port_1															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 8.48 aSercosErrorCount Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b16	aSercosErrorCount_Port_2	The counter counts Sercos III Ethernet frames that have a wrong FCS or are misaligned on port 2. This counter saturates at 0xFFFF and can be cleared by writing to it.	R
b15 to b0	aSercosErrorCount_Port_1	The counter counts Sercos III Ethernet frames that have a wrong FCS or are misaligned on port 1. This counter saturates at 0xFFFF and can be cleared by writing to it.	R

8.4.47 MSTLMAX — Maximum Sequential MST Losses

Address: 4402 00E0h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	value															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 8.49 MSTLMAX Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b16	Reserved	Reserved	R
b15 to b0	value	Amount of sequential MST losses for Interrupt 10 and 11	R/W

8.4.48 MSTLSUM — Sum MST Losses

Address: 4402 00E4h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	value															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 8.50 MSTLSUM Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b16	Reserved	Reserved	R
b15 to b0	value	Sum of sequential MST losses	R/W

8.4.49 MSTLACT — Actual Sequential MST Losses

Address: 4402 00E8h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	value															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 8.51 MSTLACT Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b16	Reserved	Reserved	R
b15 to b0	value	Actual sequential MST losses	R/W

8.4.50 TMDSCL — Timing Descriptor Lower

Address: 4402 0100h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	Event_TCNT_Value										
Value after reset	X	X	X	X	X	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	Event_TCNT_Value															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 8.52 TMDSCL Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b27	Reserved	Reserved	R
b26 to b0	Event_TCNT_Value	Nanosecond value for TCNT event	R/W

8.4.51 TMDSCU — Timing Descriptor Upper

Address: 4402 0104h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	Event_type				—	—	—	—	—	—	SCC_Select	
Value after reset	X	X	0	0	0	0	0	0	X	X	X	X	X	X	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	SCC_Value							
Value after reset	X	X	X	X	X	X	X	X	0	0	0	0	0	0	0	0

Table 8.53 TMDSCU Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b28	Reserved	Reserved	R
b27 to b24	Event_type	Type of Event	R/W
b23 to b18	Reserved	Reserved	R
b17, b16	SCC_Select	Selection of subcycle counter 0: Sercos III cycle 1: Subcycle counter A 2: Subcycle counter B 3: Cycle Counter MDT	R/W
b15 to b8	Reserved	Reserved	R
b7 to b0	SCC_Value	Subcycle count for event enable	R/W

8.4.52 TMDSCSEL — Timing Descriptor Select

Address: 4402 0108h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TMDSCSEL															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TMDSCSEL															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 8.54 TMDSCSEL Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	TMDSCSEL	Timing Descriptor Select (0 to 15)	R/W

8.4.53 PTMDSCL — Port Timing Descriptor Lower

Address: 4402 0110h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	Event_TCNT_Value										
Value after reset	X	X	X	X	X	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	Event_TCNT_Value															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 8.55 PTMDSCL Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b27	Reserved	Reserved	R
b26 to b0	Event_TCNT_Value	Nanosecond value for TCNT event	R/W

8.4.54 PTMDSCU — Port Timing Descriptor Upper

Address: 4402 0114h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	Event_type				—	—	—	—	—	—	SCC_Select	
Value after reset	X	X	X	X	0	0	0	0	X	X	X	X	X	X	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	SCC_Value							
Value after reset	X	X	X	X	X	X	X	X	0	0	0	0	0	0	0	0

Table 8.56 PTMDSCU Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b28	Reserved	Reserved	R
b27 to b24	Event_type	Type of Event	R/W
b23 to b18	Reserved	Reserved	R
b17, b16	SCC_Select	Selection of subcycle counter 0: Sercos III cycle 1: Subcycle counter A 2: Subcycle counter B 3: Cycle Counter MDT	R/W
b15 to b8	Reserved	Reserved	R
b7 to b0	SCC_Value	Subcycle count for event enable	R/W

8.4.55 PTMDSCSEL — Port Timing Descriptor Select

Address: 4402 0118h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	PTMDSCSEL															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PTMDSCSEL															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 8.57 PTMDSCSEL Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	PTMDSCSEL	Timing Descriptor Select (0 to 15)	R/W

8.4.56 STNS — System Time (nanoseconds)

Address: 4402 0130h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	Nanoseconds													
Value after reset	X	X	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	Nanoseconds															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 8.58 STNS Register Contents

Bit Position	Bit Name	Function	R/W
b31, b30	Reserved	Reserved	R
b29 to b0	Nanoseconds	Nanosecond value of system time, counts up to 1 second	R

8.4.57 STSEC — System Time (seconds)

Address: 4402 0134h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	Seconds															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	Seconds															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 8.59 STSEC Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	Seconds	Seconds of system time (Unix Time)	R

8.4.58 STNSTSR — System Time TSRef (nanoseconds)

Address: 4402 0138h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	Nanoseconds													
Value after reset	X	X	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	Nanoseconds															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 8.60 STNSTSR Register Contents

Bit Position	Bit Name	Function	R/W
b31, b30	Reserved	Reserved	R
b29 to b0	Nanoseconds	Nanosecond value of system time at last TSRef	R

8.4.59 STSECTSR — System Time TSRef (seconds)

Address: 4402 013Ch

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	Seconds															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	Seconds															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 8.61 STSECTSR Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	Seconds	Seconds of system time (Unix Time) at last TSRef	R

8.4.60 SCCAB — Subcycle Counter Control & Status

Address: 4402 0140h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	SccValueB								SccValueA							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SccMaxB								SccMaxA							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 8.62 SCCAB Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b24	SccValueB	Subcycle value of sub cycle counter B	R
b23 to b16	SccValueA	Subcycle value of sub cycle counter A	R
b15 to b8	SccMaxB	Maximum value for sub cycle counter B	R/W
b7 to b0	SccMaxA	Maximum value for sub cycle counter A	R/W

8.4.61 SCCMDT — Subcycle Counter MDT

Address: 4402 0150h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	Actual_counter													
Value after reset	X	X	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	Maximum_count													
Value after reset	X	X	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 8.63 SCCMDT Register Contents

Bit Position	Bit Name	Function	R/W
b31, b30	Reserved	Reserved	R
b29 to b16	Actual_counter	Actual value active in timing processor	R
b15, b14	Reserved	Reserved	R
b13 to b0	Maximum_count	Internal counter is reset when it exceeds this value or cycle counter in MDT0 is zero.	R/W

8.4.62 RXBUF[n]_P1A — Receive Buffer [n] Base Address for Port 1 and Buffer System A (n = 0..2)

Address: 4402 0180h +4h × n

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	RXBUF[n]_P1A															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RXBUF[n]_P1A															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 8.64 RXBUF[n]_P1A Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	RXBUF[n]_P1A	Receive buffer [n] base address for port 1 and buffer system A	R/W

8.4.63 RXBUF[n]_P1B — Receive Buffer [n] Base Address for Port 1 and Buffer System B (n = 0..2)

Address: 4402 0190h +4h × n

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	RXBUF[n]_P1B															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RXBUF[n]_P1B															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 8.65 RXBUF[n]_P1B Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	RXBUF[n]_P1B	Receive buffer [n] base address for port 1 and buffer system B	R/W

8.4.64 RXBUF_P1SVC — Receive Buffer Base Address for Service Channel Data at Port 1

Address: 4402 019Ch

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	RXBUF_P1SVC															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RXBUF_P1SVC															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 8.66 RXBUF_P1SVC Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	RXBUF_P1SVC	Receive buffer base address for service channel data at port 1	R/W

8.4.65 RXBUF[n]_P2A — Receive Buffer [n] Base Address for Port 2 and Buffer System A (n = 0..2)

Address: 4402 01A0h +4h × n

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	RXBUF[n]_P2A															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RXBUF[n]_P2A															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 8.67 RXBUF[n]_P2A Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	RXBUF[n]_P2A	Receive buffer [n] base address for port 2 and buffer system A	R/W

8.4.66 RXBUF[n]_P2B — Receive Buffer [n] Base Address for Port 2 and Buffer System B (n = 0..2)

Address: 4402 01B0h +4h × n

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	RXBUF[n]_P2B															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RXBUF[n]_P2B															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 8.68 RXBUF[n]_P2B Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	RXBUF[n]_P2B	Receive buffer [n] base address for port 2 and buffer system B	R/W

8.4.67 RXBUF_P2SVC — Receive Buffer Base Address for Service Channel Data at Port 2

Address: 4402 01BCh

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	RXBUF_P2SVC															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RXBUF_P2SVC															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 8.69 RXBUF_P2SVC Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	RXBUF_P2SVC	Receive buffer base address for service channel data at port 2	R/W

8.4.68 TXBUF[n]_A — Transmit Buffer [n] Base Address for Buffer System A (n = 0..3)

Address: 4402 01C0h +4h × n

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TXBUF[n]_A															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TXBUF[n]_A															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 8.70 TXBUF[n]_A Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	TXBUF[n]_A	Transmit buffer [n] base address for buffer system A	R/W

8.4.69 TXBUF[n]_B — Transmit Buffer [n] Base Address for Buffer System B (n = 0..3)

Address: 4402 01D0h +4h × n

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TXBUF[n]_B															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TXBUF[n]_B															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 8.71 TXBUF[n]_B Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	TXBUF[n]_B	Transmit buffer [n] base address for buffer system B	R/W

8.4.70 TXBUF_P1 — Transmit Buffer Base Address for Port 1 Only

Address: 4402 01F0h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TXBUF_P1															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TXBUF_P1															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 8.72 TXBUF_P1 Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	TXBUF_P1	Transmit buffer base address for port 1 only	R/W

8.4.71 TXBUF_P2 — Transmit Buffer Base Address for Port 2 Only

Address: 4402 01F4h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TXBUF_P2															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TXBUF_P2															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 8.73 TXBUF_P2 Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	TXBUF_P2	Transmit buffer base address for port 2 only	R/W

8.4.72 TXBUF_SVC — Transmit Buffer Base Address for Service Channel Data

Address: 4402 01FCh

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TXBUF_SVC															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TXBUF_SVC															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 8.74 TXBUF_SVC Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	TXBUF_SVC	Transmit buffer base address for service channel data	R/W

8.4.73 RXBUFCSR_A — Receive Buffer Control Buffer System A

Address: 4402 0200h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	RxRequest	—	Port_2_Overflow	Port_2_New_Data	—	—	RxBuf_Port_2		—	—	Port_1_Overflow	Port_1_New_Data	—	—	RxBuf_Port_1	
Value after reset	1	X	0	0	X	X	0	1	X	X	0	0	X	X	0	1

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Buffer_Count	
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	1	0

Table 8.75 RXBUFCSR_A Register Contents

Bit Position	Bit Name	Function	R/W
b31	RxRequest	buffer system is usable (read) / request newest receive buffers (write)	R/W
b30	Reserved	Reserved	R
b29	Port_2_Overflow	data overflow of port 2 between last two receive buffer requests	R
b28	Port_2_New_Data	actual receive buffer for port 2 contains new valid data	R
b27, b26	Reserved	Reserved	R
b25, b24	RxBuf_Port_2	actual receive buffer for port 2 (system buffer)	R
b23, b22	Reserved	Reserved	R
b21	Port_1_Overflow	data overflow of port 1 between last two receive buffer requests	R
b20	Port_1_New_Data	actual receive buffer for port 1 contains new valid data	R
b19, b18	Reserved	Reserved	R
b17, b16	RxBuf_Port_1	actual receive buffer for port 1 (system buffer)	R
b15 to b2	Reserved	Reserved	R
b1, b0	Buffer_Count	00b: single buffer system 01b: double buffer system 10b: triple buffer system 11b: not defined	R/W

8.4.74 RXBUFTV_A — Rx Buffer Telegram Valid A

Address: 4402 0204h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	Port_2_AT3	Port_2_AT2	Port_2_AT1	Port_2_AT0	—	—	—	—	Port_2_MDT3	Port_2_MDT2	Port_2_MDT1	Port_2_MDT0
Value after reset	X	X	X	X	0	0	0	0	X	X	X	X	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	Port_1_AT3	Port_1_AT2	Port_1_AT1	Port_1_AT0	—	—	—	—	Port_1_MDT3	Port_1_MDT2	Port_1_MDT1	Port_1_MDT0
Value after reset	X	X	X	X	0	0	0	0	X	X	X	X	0	0	0	0

Table 8.76 RXBUFTV_A Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b28	Reserved	Reserved	R
b27	Port_2_AT3	Buffer contains valid data of AT3 from port 2	R
b26	Port_2_AT2	Buffer contains valid data of AT2 from port 2	R
b25	Port_2_AT1	Buffer contains valid data of AT1 from port 2	R
b24	Port_2_AT0	Buffer contains valid data of AT0 from port 2	R
b23 to b20	Reserved	Reserved	R
b19	Port_2_MDT3	Buffer contains valid data of MDT3 from port 2	R
b18	Port_2_MDT2	Buffer contains valid data of MDT2 from port 2	R
b17	Port_2_MDT1	Buffer contains valid data of MDT1 from port 2	R
b16	Port_2_MDT0	Buffer contains valid data of MDT0 from Port 2	R
b15 to b12	Reserved	Reserved	R
b11	Port_1_AT3	Buffer contains valid data of AT3 from port 1	R
b10	Port_1_AT2	Buffer contains valid data of AT2 from port 1	R
b9	Port_1_AT1	Buffer contains valid data of AT1 from port 1	R
b8	Port_1_AT0	Buffer contains valid data of AT0 from port 1	R
b7 to b4	Reserved	Reserved	R
b3	Port_1_MDT3	Buffer contains valid data of MDT3 from port 1	R
b2	Port_1_MDT2	Buffer contains valid data of MDT2 from port 1	R
b1	Port_1_MDT1	Buffer contains valid data of MDT1 from port 1	R
b0	Port_1_MDT0	Buffer contains valid data of MDT0 from Port 1	R

8.4.75 RXBUFTR_A — Rx Buffer Telegram Requirements A

Address: 4402 0208h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	AT3_required	AT2_required	AT1_required	AT0_required	—	—	—	—	MDT3_required	MDT2_required	MDT1_required	MDT0_required
Value after reset	X	X	X	X	0	0	0	0	X	X	X	X	0	0	0	0

Table 8.77 RXBUFTR_A Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b12	Reserved	Reserved	R
b11	AT3_required	Buffer change only when AT3 was valid	R/W
b10	AT2_required	Buffer change only when AT2 was valid	R/W
b9	AT1_required	Buffer change only when AT1 was valid	R/W
b8	AT0_required	Buffer change only when AT0 was valid	R/W
b7 to b4	Reserved	Reserved	R
b3	MDT3_required	Buffer change only when MDT3 was valid	R/W
b2	MDT2_required	Buffer change only when MDT2 was valid	R/W
b1	MDT1_required	Buffer change only when MDT1 was valid	R/W
b0	MDT0_required	Buffer change only when MDT0 was valid	R/W

8.4.76 TXBUFCSR_A — Transmit Buffer Control for Buffer System A

Address: 4402 020Ch

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TxRequest	—	—	—	—	—	—	—	—	—	—	—	—	—	TxBuf	
Value after reset	1	X	X	X	X	X	X	X	X	X	X	X	X	X	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Buffer_Count	
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	1	1

Table 8.78 TXBUFCSR_A Register Contents

Bit Position	Bit Name	Function	R/W
b31	TxRequest	buffer system is usable (read) / request newest transmit buffer (write)	R/W
b30 to b18	Reserved	Reserved	R
b17, b16	TxBuf	actual transmit buffer (system buffer)	R/W
b15 to b2	Reserved	Reserved	R
b1, b0	Buffer_Count	00b: single buffer system 01b: double buffer system 10b: triple buffer system 11b: quad buffer system	R/W

8.4.77 RXBUFCSR_B — Receive Buffer Control Buffer System B

Address: 4402 0210h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	RxRequest	—	Port_2_Overflow	Port_2_New_Data	—	—	RxBuf_Port_2		—	—	Port_1_Overflow	Port_1_New_Data	—	—	RxBuf_Port_1	
Value after reset	1	X	0	0	X	X	0	1	X	X	0	0	X	X	0	1

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Buffer_Count	
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	1	1

Table 8.79 RXBUFCSR_B Register Contents

Bit Position	Bit Name	Function	R/W
b31	RxRequest	buffer system is usable (read) / request newest transmit buffer (write)	R/W
b30	Reserved	Reserved	R
b29	Port_2_Overflow	data overflow of port 2 between last two receive buffer requests	R
b28	Port_2_New_Data	actual receive buffer for port 2 contains new valid data	R
b27, b26	Reserved	Reserved	R
b25, b24	RxBuf_Port_2	actual receive buffer for port 2 (system buffer)	R
b23, b22	Reserved	Reserved	R
b21	Port_1_Overflow	data overflow of port 1 between last two receive buffer requests	R
b20	Port_1_New_Data	actual receive buffer for port 1 contains new valid data	R
b19, b18	Reserved	Reserved	R
b17, b16	RxBuf_Port_1	actual receive buffer for port 1 (system buffer)	R
b15 to b2	Reserved	Reserved	R
b1, b0	Buffer_Count	00b: single buffer system 01b: double buffer system 10b: triple buffer system 11b: not defined	R/W

8.4.78 RXBUFTV_B — Rx Buffer Telegram Valid B

Address: 4402 0214h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	Port_2_AT3	Port_2_AT2	Port_2_AT1	Port_2_AT0	—	—	—	—	Port_2_MDT3	Port_2_MDT2	Port_2_MDT1	Port_2_MDT0
Value after reset	X	X	X	X	0	0	0	0	X	X	X	X	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	Port_1_AT3	Port_1_AT2	Port_1_AT1	Port_1_AT0	—	—	—	—	Port_1_MDT3	Port_1_MDT2	Port_1_MDT1	Port_1_MDT0
Value after reset	X	X	X	X	0	0	0	0	X	X	X	X	0	0	0	0

Table 8.80 RXBUFTV_B Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b28	Reserved	Reserved	R
b27	Port_2_AT3	Buffer contains valid data of AT3 from port 2	R
b26	Port_2_AT2	Buffer contains valid data of AT2 from port 2	R
b25	Port_2_AT1	Buffer contains valid data of AT1 from port 2	R
b24	Port_2_AT0	Buffer contains valid data of AT0 from port 2	R
b23 to b20	Reserved	Reserved	R
b19	Port_2_MDT3	Buffer contains valid data of MDT3 from port 2	R
b18	Port_2_MDT2	Buffer contains valid data of MDT2 from port 2	R
b17	Port_2_MDT1	Buffer contains valid data of MDT1 from port 2	R
b16	Port_2_MDT0	Buffer contains valid data of MDT0 from Port 2	R
b15 to b12	Reserved	Reserved	R
b11	Port_1_AT3	Buffer contains valid data of AT3 from port 1	R
b10	Port_1_AT2	Buffer contains valid data of AT2 from port 1	R
b9	Port_1_AT1	Buffer contains valid data of AT1 from port 1	R
b8	Port_1_AT0	Buffer contains valid data of AT0 from port 1	R
b7 to b4	Reserved	Reserved	R
b3	Port_1_MDT3	Buffer contains valid data of MDT3 from port 1	R
b2	Port_1_MDT2	Buffer contains valid data of MDT2 from port 1	R
b1	Port_1_MDT1	Buffer contains valid data of MDT1 from port 1	R
b0	Port_1_MDT0	Buffer contains valid data of MDT0 from Port 1	R

8.4.79 RXBUFTR_B — Rx Buffer Telegram Requirements B

Address: 4402 0218h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	AT3_required	AT2_required	AT1_required	AT0_required	—	—	—	—	MDT3_required	MDT2_required	MDT1_required	MDT0_required
Value after reset	X	X	X	X	0	0	0	0	X	X	X	X	0	0	0	0

Table 8.81 RXBUFTR_B Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b12	Reserved	Reserved	R
b11	AT3_required	Buffer change only when AT3 was valid	R/W
b10	AT2_required	Buffer change only when AT2 was valid	R/W
b9	AT1_required	Buffer change only when AT1 was valid	R/W
b8	AT0_required	Buffer change only when AT0 was valid	R/W
b7 to b4	Reserved	Reserved	R
b3	MDT3_required	Buffer change only when MDT3 was valid	R/W
b2	MDT2_required	Buffer change only when MDT2 was valid	R/W
b1	MDT1_required	Buffer change only when MDT1 was valid	R/W
b0	MDT0_required	Buffer change only when MDT0 was valid	R/W

8.4.80 TXBUFCSR_B — Transmit Buffer Control for Buffer System B

Address: 4402 021Ch

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TxRequest	—	—	—	—	—	—	—	—	—	—	—	—	—	TxBuf	
Value after reset	1	X	X	X	X	X	X	X	X	X	X	X	X	X	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Buffer_Count	
Value after reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	1	1

Table 8.82 TXBUFCSR_B Register Contents

Bit Position	Bit Name	Function	R/W
b31	TxRequest	buffer system is usable (read) / request newest transmit buffer (write)	R/W
b30 to b18	Reserved	Reserved	R
b17, b16	TxBuf	actual transmit buffer (system buffer)	R/W
b15 to b2	Reserved	Reserved	R
b1, b0	Buffer_Count	00b: single buffer system 01b: double buffer system 10b: triple buffer system 11b: quad buffer system	R/W

8.5 Operation

8.5.1 Initializing Flow

8.5.1.1 Initializing

The initialization sequence in this section is a example for preparation of system environments for using SERCOS3 under configuration below.

SERCOS3 Configuration of This Example:

- 2 ports are connected to external port through MII Converter.

Note 1. RZ/N1 User's Manual : System Introduction, Multiplexing, Electrical and Mechanical Information

Note 2. Ethernet Accessory Register

Note 3. Sercos III Slave Controller

Figure 8.2 Initializing Flowchart

8.5.1.2 SERCOS3_IP_INIT

SERCOS3_IP_INIT sequence is used for initializing of SERCOS3 module.

For SERCOS3_IP_INIT operation, complete the following flowchart:

Figure 8.3 SERCOS3 IP Initialization Flowchart

8.5.1.3 SERCOS3_DSC_INIT

SERCOS3_SDC_INIT sequence is used for initializing of Descriptor of SERCOS3 module.

For SERCOS3_SDC_INIT operation, complete the following flowchart:

Figure 8.4 SERCOS3_DSC_INIT Flowchart

8.5.1.4 SERCOS3_TCNT_INIT_START

SERCOS3_TCNT_INIT_START sequence is used for initializing and start of Timing Descriptor and Port Timing Descriptor of SERCOS3 module.

For SERCOS3_TCNT_INIT_START operation, complete the following flowchart:

Figure 8.5 SERCOS3_TCNT_INIT_START Flowchart

8.5.2 Communication Flow

8.5.2.1 Transmit

In RT state, this Sercos module receives periodically the AT frames from Sercos III master, this module sends the prepared data in Tx RAM automatically.

In NRT state, this Sercos module sends the prepared data in Tx RAM by the indication of software.

8.5.2.2 Receive

In RT state, this Sercos module receives periodically the MDT frame and AT frames from Sercos III master, and stores these data to Rx RAM.

In NRT state, this Sercos module receives the Ethernet frame (IP frame), and stores these data to Rx RAM.

Figure 8.6 Memory Map in SERCOS III

CAUTION

SVC RAM can't be accessed with 8bit width.

8.5.3 Interrupt Control

For controlling of interrupts three registers are implemented:

Table 8.83 Interrupt Map for Slave Device

Address	Register Symbol	Register Name
4402 0008h	IER0	Interrupt Enable Register
4402 0010h	IMR0	Interrupt Multiplex Register
4402 0018h	IRR0	Interrupt Reset/Status Register

(1) Interrupt Map for Slave Device

The slave device spends two hardware interrupt outputs SERCOS3_Int0 and SERCOS3_Int1 and supports up to thirty two interrupt sources.

Table 8.84 Interrupt Map for Slave Device

Interrupt Number	Interrupt Source	
Interrupt 0	Int_TINT[0]	Event TINT[0], created by timer/counter TCNT
Interrupt 1	Int_TINT[1]	Event TINT[1], created by timer/counter TCNT
Interrupt 2	Int_TINT[2]	Event TINT[2], created by timer/counter TCNT
Interrupt 3	Int_TINT[3]	Event TINT[3], created by timer/counter TCNT
Interrupt 4	Int_TINTMAX	Event TMAX, created by timer/counter TCNT
Interrupt 5	Int_MDTx/ATx-Valid	Interrupt after valid user defined Sercos III frame (SFCR)
Interrupt 6	Int_MDTx/ATx-Valid	Interrupt after valid user defined Sercos III frame (SFCR)
Interrupt 7	Int_DIVCLK	Interrupt from DIV_CLK unit
Interrupt 8	Int_IPIntPort1	Event IP port 1
Interrupt 9	Int_IPIntPort2	Event IP port 2
Interrupt 10	Int_Half_MST_error	((Maximum count + 1) >> 1) of sequential MST errors reached
Interrupt 11	Int_MST_error	Maximum count of sequential MST errors reached
Interrupt 12	Int_RxBufReqPort1	Internal reception buffer change for port 1
Interrupt 13	Int_RxBufReqPort2	Internal reception buffer change for port 2
Interrupt 14-15	—	Not used
Interrupt 16-23	Int_SVC[7:0]	Service channel interrupt 0 - 7
Interrupt 24-31	—	Not used

(a) Interrupt Enable Register (IER0)

Each interrupt source can be separately enabled. An interrupt source is enabled by setting the corresponding IE[n] bit to ONE.

After power up or reset, all bits of the register are set to ZERO.

(b) Interrupt Multiplex Register (IMR0)

Each interrupt source can be mapped separately to one output. The multiplex bits assign the enabled interrupt sources to one of the two hardware outputs SERCOS3_Int0 or SERCOS3_Int1. When the corresponding bit is set to ZERO, the interrupt source is assigned to SERCOS3_Int0, when set to ONE, the interrupt source is assigned to SERCOS3_Int1.

(c) Interrupt Reset/Status Register (IRR0)

The actual state of the enabled interrupt source can be sampled by the Interrupt Status register IRR0. A logical ONE in the corresponding bit means the source IR[n] is active. To clear an interrupt, the corresponding bit has to be written with a ONE.

8.5.4 Slave Timing Control

The slave timing control is based on three independent timers.

- TCNT: Central timer for system timing
- TCNT[1]: Timer for communication specific events on port 1
- TCNT[2]: Timer for communication specific events on port 2

Figure 8.7 Slave Timing Control

All timers and related registers have a resolution of 1 ns, although the timers are incremented only in 20 ns steps. The derived events are assigned by 16 event registers. All timers are controlled (enabled) by the TCSR register.

8.5.4.1 Timing Control and Status Register (TCSR)

The timing control and status register is implemented to setup the counter functionality.

For this purpose, the register controls

- the enabling of the counter TCNT, TCNT 1/2,
- the CON_CLK output enable and polarity,
- the DIV_CLK functionality by mode and polarity,
- which Port in line is used for synchronization (MST disable)

Table 8.85 Timing Control and Status Register

Address	Register Symbol	Register Name
4402 0038h	TCSR	Timing Control/Status Register

The disabling of MSTs for synchronization is done automatically in line topology. This module enables only the port for synchronization where the MST is received first. In Ring topology, no automatism is implemented. The firmware has to enable both ports as soon as synchronization delays for both ports are valid. The validness of the synchronization delays cannot be decided by logic.

8.5.4.2 System Timer Read Back Register (STRBR)

The actual state of the TCNT counter. Time resolution LSB is 1 ns.

Table 8.86 System Timer Read Back Register

Address	Register Symbol	Register Name
4402 0034h	STRBR	System Timer Read Back Register

8.5.4.3 Central Timer TCNT

The event register are addressed indirectly over a select register. The user can define up to 16 timing events from index 0 to 15. The timing descriptors are read only and the select register (108h) is ignored when TCNT is enabled (see ET0 of TCSR).

Table 8.87 TCNT Event Register Map

Address	Register Symbol	Register Name
4402 0100h	TMDSCL	Timing Descriptor Lower
4402 0104h	TMDSCU	Timing Descriptor Upper
4402 0108h	TMDSCSEL	Timing Descriptor Select

Table 8.88 Event Types of Timing Descriptor (TMDSCU)

Event Type	
0	No Event
1	Event TINT[0]
2	Event TINT[1]
3	Event TINT[2]
4	Event TINT[3]
5	Sync Set
6	Sync Reset
7	(Not used)
8	(Not used)
9	(Not used)
10	(Not used)
11	(Not used)
12	Reload
13	Reload value

(1) TCNT Functionality and Programming Events

TCNT is a 26-bit up counter with 1 ns register- and 20 ns time resolution. It is the central timer for controlling the system timing. For this purpose, TCNT can create six different events and the hardware signal S3_CONCLK.

The hardware signal S3_CONCLK is set with “Sync Set” and cleared with “Sync Reset”.

The counter is cleared with a TMST event, followed by ramping in 20 ns steps until the next TMST is recognized or Reload event (overrun) occurs. An overrun happens when the TMST was missed. With the TMST, the first event memory location is selected.

During the ramping up phase, the counter is compared with the actual selected event memory location. If the counter is greater or equal the event time (Event_TCNT_Value), an event corresponding to the programmed event type is generated and the state machine selects the following event memory location.

Special care must be taken to operate Reload event, this event is always the last event in the event table. When TCNT reaches Reload event, TCNT is reloaded with the value TMAX located in the event memory location following Reload event.

The Reload event has to be at least 200 ppm greater than the cycle time to compensate runtime differences of the oscillators. The input jitter of the TMST has to be mentioned additionally.

Example)

An event structure is created by following event memory assignment:

Table 8.89 Example of Event Structure

Event Address	Event
0x00	Sync Set
0x01	Sync Reset
0x02	Event TINT[0]
0x03	Reload
0x04	Reload Value

Figure 8.8 Example for TCNT Timing

8.5.4.4 Communication Timers TCNT[Port 1 / Port 2]

Parallel to the central timer unit TCNT the slave device has two communication line related timers TCNT[1] and TCNT[2] for controlling and monitoring Ethernet travel.

The event register are addressed indirectly over a select register. The user can define up to 16 timing events from index 0 to 15. The timing descriptors are read only and the select register (118h) is ignored when TCNT1/2 is enabled (see ET1_2 of TCSR) and CPS bit in PHASESR is not set. During CPS, the event list is stored into a second list, which remains active for the ports. The new list is overtaken when CPS is reset or TCNT1/2 is disabled. This way, the firmware can prepare the list for the following communication phase without disabling TCNT1/2 and without loss of NRT channel during CPS.

Table 8.90 TCNT1/2 Event Register Map

Address	Register Symbol	Register Name
4402 0110h	PTMDSCL	Port Timing Descriptor Lower
4402 0114h	PTMDSCU	Port Timing Descriptor Upper
4402 0118h	PTMDSCSEL	Port Timing Descriptor Select

(1) TCNT[1/2] Functionality and Programming Events

TCNT[1] and TCNT[2] are 27-bit up counters with 1 ns register- and 40 ns time resolution. They are the communication timers for controlling and monitoring Ethernet telegram travel.

Table 8.91 TCNT[1/2] Functionality and Programming Events

Code	Event	Description
1	Event_IPChannel_Open	Opens the IP-channel, enabling the MAC for transmission of NRT Ethernet frames. Switches from fast-forward to store and forward.
2	Event_IPChannel_TxClose	Defines the latest transmission point for Ethernet frames inside the IP-channel to avoid violations. The Tx MAC disables the transmission until the next IP timeslot.
3	Event_IPChannel_RxClose	Closes the IP-channel. Switches from store and forward to fast-forward.
4	Event_AT0WindowOpen	Opens the AT0 monitoring window.
5	Event_AT0WindowClose	Closes the AT0 monitoring window
6	Event_RxBufRequest_BufSysA	Performs a receive buffer change of buffer system A
7	Event_RxBufRequest_BufSysB	Performs a receive buffer change of buffer system B
8	Event_TxBufRequest_BufSysA	Performs a transmit buffer change of buffer system A
9	Event_TxBufRequest_BufSysB	Performs a transmit buffer change of buffer system B
10	Event_MSTWindowOpen	Opens the MST monitoring window, MST telegrams received before this event will create a MST window error. Receiving a MST also closes this window.
11	Event_MSTWindowClose	Closes the MST monitoring window and creates an MST window Error. Overrun event, if the timer isn't reset by a valid MST[1] or MST[2].
12	Reload Value	Counter is reloaded with this value after Event_MSTWindowClose
13	Event_SVCStart	Trigger service channel processor.
14	Event_MSTHeaderWindowOpen	Opens MST header window for broadcast reduction of loopback slave. Place this event 2.24 µs before opening of MST window.

Counter TCNT[1] is cleared with an TMST[1] event, counter TCNT[2] is cleared with an TMST[2] event, followed by ramping in 40 ns steps until the next TMST[1/2] is recognized or a MSTWindow Close event (overrun) occurs. An overrun happens when the TMST[1/2] is missed or is out of the maximum receiving window.

During the ramping up phase, the counters are compared with the actual selected event memory location for TCNT[1] or TCNT[2]. If the counter is greater or equal the event time (Event_TCNT_Value), an event corresponding to the programmed event type is generated and the state machine for TCNT[1] or TCNT[2] selects the following event memory location.

Special care must be taken to operate MSTWindowClose events, these events are always the last event in the event table. When TCNT[1] or TCNT[2] reaches MSTWindowClose, TCNT[1] or TCNT[2] is reloaded with the value TMAX located in the event memory location following MSTWindowClose. The MST Window Close event has to be at least 200 ppm greater than the cycle time to compensate runtime differences of the oscillators. The input jitter of the arriving MST has to be mentioned additionally. Next figure shows a typical example for TCNT[1].

Example)

Table 8.92 Example of TCNT[1/2] Event Structure

Event Address	Event
00h	Event_RxBufRequest_BufSysA
01h	Event_IPChannel_Open
02h	Event_IPChannel_TxClose
03h	Event_IPChannel_RxClose
04h	Event_AT0WindowOpen
05h	Event_AT0WindowClose
06h	Event_MSTWindowOpen
07h	Event_MSTWindowClose
08h	Reload Value

Figure 8.9 Example for TCNT[1/2] Timing

8.5.4.5 System Timer

Table 8.93 System Timer Register Map

Address	Register Symbol	Register Name
4402 0130h	STNS	System Time (nanoseconds)
4402 0134h	STSEC	System Time (seconds)
4402 0138h	STNSTSR	System Time TSRef (nanoseconds)
4402 013Ch	STSECTSR	System Time TSRef (seconds)

(1) System Time Nano Seconds (STNS)

Temporary value of system time nanoseconds. When ET3 of TCSR is enabled, every write access on this register updates read contents of STNS and STSEC. Otherwise system time can be written by firmware.

(2) Seconds (STSEC)

Temporary value of system time seconds. Will be updated by every write access on system time nanoseconds (STNS , 0x130) while ET3 of TCSR is enabled. Otherwise seconds are writeable by firmware.

(3) Nanoseconds TSRef (STNSTSR)

The internal nanoseconds of the system time are stored automatically at TSRef (TMST with jitter) and readable through this register.

(4) Seconds TSRef (STSECTSR)

The internal nanoseconds of the system time are stored automatically at TSRef (TMST with jitter) and readable through this register.

8.5.4.6 Subcycle Counters

Table 8.94 Subcycle Counters Register Map

Address	Register Symbol	Register Name
4402 0140h	SCCAB	Subcycle Counter Control & Status
4402 0150h	SCCMDT	Subcycle Counter MDT

(1) Subcycle Counters (SCCAB)

Subcycle counters used for timing events to not become active in each Sercos III cycle.

(2) Sybcycle counter (SCCMDT)

Synchronization counter for all sub cycle counter (producer cycles). When this counter has a value of 0, all other counters are also reset to 0.

Example)

Following image gives an easy example to realize sub cycles. Event TINT[0] is programmed with a “Sub cycle counter select” value of 1 to select sub cycle counter A and becomes active when this counter has a value of 0 (“Sub cycle count value”). The second event TINT[1] is programmed with a “Sub cycle counter select” value of 2 to select sub cycle counter B and becomes also active when this counter has a value of 1 (“Sub cycle count value”). The maximum values of each sub cycle counter is written into SCCAB (140h). In this example, SccMaxA is 1 and SccMaxB is 2. The least common multiple of both sub cycles is 6, because of the additional 0 value. So the maximum value for SCCMDT (150h) is set to 5.

Figure 8.10 Example for Sub Cycle Counters

8.5.5 Slave MST Processing

The slave device receives on port 1 and port 2 MST telegrams, which are used to create a central TMST signal after a defined delay, and to measure the ring delay. This is done by the following unit.

Figure 8.11 Slave MST Processing

Two 19-bit up counters, one for port 1 and one for port 2, are reset by TMST[1] and TMST[2] events. The counters and related registers have 1 ns resolution but are incremented in 40 ns steps. Maximum operating time is therefore around 524 µs. After a reset, the counters ramp up until they match the timing delays programmed in the time delay register TDMST1 or time delay register TDMST2. With a match, the TMST signal is triggered.

During the ramp up phase, both counters are added and compared to a dedicated ring delay, programmed in the ring delay register TRDLY. When the comparator inputs match the actual states of counter, MST[Port 1] and MST[Port 2] are captured in the Sync Time registers.

8.5.5.1 Register Interface for Slave MST Processing

Table 8.95 Slave MST Processing Register Map

Address	Register Symbol	Register Name
4402 003Ch	TRDLY	Ring Delay Register
4402 0040h	TDMST1	Time Delay MST Port 1
4402 0044h	TDMST2	Time Delay MST Port 2
4402 0048h	SCR1	Sync Time Register Port 1
4402 004Ch	SCR2	Sync Time Register Port 2

(1) Ring delay Register (TRDLY)

The user must program the delay that the master has measured into the ring delay register. This time is used to calculate the delays from each port to the master.

(2) Sync Count Register (SCR1/2)

After the user program has written the ring delay, the logic calculates the delay values from each port to the master and writes it into the Sync Count registers. The Calculation starts with the first arriving MST and takes at least Ring delay in time to complete. This process runs the whole time when MST frames are arriving.

(3) Time Delay MST Register (TDMST1/2)

The main counter TCNT is synchronized by the TMST signal. This signal is generated from both MST signals in ring mode and from one MST in line mode. The TDMST registers are used to shift the incoming MST signals to the same point in time. TDMST1 is used to delay the MST from port 1 and TDMST2 is used to delay the MST from port 2. The delayed MSTs are combined (ORed) to form the TMST signal (first rules). The TMST signal is then used to reset the TCNT counter to 0 for synchronization.

8.5.6 DIV_CLK Function

To couple communication and control task timing, it's advantageous for many applications to provide DIV_CLK functionality.

Table 8.96 DIV_CLK Function Register Map

Address	Register Symbol	Register Name
4402 0054h	DTDIVCLK	Delay Time for DIV_CLK
4402 0058h	TDIV_NDIVCLK	DIV_CLK Time / Count Register

Two DIV_CLK modes are available:

Table 8.97 DIV_CLK Modes

	DIV_CLK Modes	
DivClk_mode	0	DIV_CLK becomes active n-times within one communication cycle
	1	DIV_CLK becomes active after n communication cycles

The DIV_CLK mode and the polarity of the DIV_CLK output are set in the timing control and status register TCSR. See description of TCSR for reference.

Pulse width of the active DIV_CLK output is fix 1 µs.

8.5.6.1 DIV_CLK mode 0 — n Times Within One Communication Cycle

When DIV_CLK mode 0 is set, DIV_CLK signal becomes active several times within a communication cycle. The delay time for the first pulse and distance between two pulses can be programmed by the registers TDIVCLK and DTDIVCLK.

Figure 8.12 TDIVCLK and DTDIVCLK in DIV_CLK mode 0

NDIVCLK controls the number of pulses within a communication cycle. When NDIVCLK is programmed to ZERO, signal DIV_CLK never become active.

8.5.6.2 DIV_CLK Mode 1 — Once after N Communication Cycle

When DIV_CLK mode 1 is set, DIV_CLK signal becomes active once after n communication cycles. The delay time for the pulse after Cyc_Start and the number of communication cycles are programmed in the register TDIVCLK/NDIVCLK.

Figure 8.13 TDIVCLK and DTDIVCLK in DIV_CLK mode 1

8.5.7 Data Flow Control

Table 8.98 Data Flow Control Register Map

Address	Register Symbol	Register Name
4402 0020h	DFCSR	Data Flow Control/Status Register
4402 0024h	PHASESR	Phase Status Register
4402 0028h	TGSR1	Telegram Reset/Status Register Port 1
4402 002Ch	TGSR2	Telegram Reset/Status Register Port 2
4402 0030h	DESCR	Descriptor Control Register
4402 007Ch	SEQCNT	Sequence Counter

8.5.7.1 Data Flow Control and Status Register (DFCSR)

This register is combining the control information for the data flow and the status information of the physical layer.

The topology changes automatically if there is a line error at the master distant port in line topology or at any port in ring topology.

8.5.7.2 Descriptor Control Register (DESCR)

To ensure that the offset to the index table is 32-bit aligned, the lower two bits of both pointers in DESCR are unused and not evaluated.

(1) Descriptor Index Table Entry

To ensure that the offset inside an index table entry is 32-bit aligned, the lower two bits are overlaid with enabling of entry (LSB = Enable). The Descriptor index table is placed inside Rx- and Tx-RAM with start address defined in Register DESCR (030h).

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	Operation
		Offset of descriptor list in RAM (32-bit aligned)													Enable	Read/Write

D31	D30	D29	D28	D27	D26	D25	D24	D23	D22	D21	D20	D19	D18	D17	D16	
																Read/Write

Table 8.99 Descriptor Index Table Entry Format

Name	Bits	Size	Description
Enable	0	1	Enable of descriptor list
Offset of descriptor list	2-13	10	Offset of descriptor list in RAM (32-bit aligned)

8.5.7.3 Sequence Counter Register (SEQCNT)

In CP0, the slave determines its position inside a Sercos III frame over the sequence counter field after the MST header. To facilitate this progress, this module automatically extracts this field from the telegram and stores it into following register. A Write access to this register resets the contents to 0x81FF. The value of an inactive port is automatically reset to 0x81FF. The firmware can easily compare both values, select the smaller one, mask out the highest bit, multiply by 2 and use it as offset for AT0 in CP0.

This register is updated as long as Rx descriptors are activated and MST phase is CP0 without CPS bit set or MST phase is CP1 with CPS bit set. This module only increments the sequence counter with the value set in DFCSR, if the Tx descriptors are activated additionally to the previous requirements.

8.5.7.4 Telegram Status Register (TGSR1/TGSR2)

The telegram monitor in the Rx processor also monitors the incoming Sercos III Ethernet telegrams on both ports.

In this context the monitor detects:

- Valid MDT0 to MDT3
- Valid AT0 to AT3
- Valid MST
- A MST out of the expected MST time window
- Missed MST
- Double missed MST, two MST are missed in sequence
- Missed AT0 header

All this information is visible in the telegram status register 1 and 2 corresponding port 1 and port 2.

The telegram status bits can be cleared by writing a ONE to the following bit locations:

Table 8.100 Clear Telegram Status Bits (TGSR1/2)

Bit	Clear
0	All MDT Bits
4	All AT Bits
8	MST Valid Bit
10	MST Window Error Bit
11	MST Miss Bit
12	MST Double Miss Bit
13	AT0 Miss Bit

(1) Bit Modification Times

Following figure is showing the times when the bits from TGSR registers are refreshed by this module.

Figure 8.14 TGSR Bit Modification Times

8.5.7.5 Supervising the Communication Phase — Phase Status Register (PHASESR)

Bits 0 to 3 and bit 7 are extracted out the Sercos III phase field of the last valid received MST field in an MDT0 frame.

8.5.8 Tx MAC Operation

8.5.8.1 Principle Structure of Tx MAC

The slave Tx MAC is able to insert data out of the transmit RAM (Tx RAM) into defined positions in an AT Ethernet telegram data stream.

To control the position and amount of data to insert, up to sixteen 32-bit descriptors are set to

- the AT telegram number (AT0 ... AT3),
- the start- and end position in the AT telegram stream
- the data type (service channel- or real time data).

8.5.8.2 Sources of Telegram Data in AT

Next figure shows the data sources for a Sercos III AT telegram.

Figure 8.15 Data Sources for a Sercos III AT Telegram

8.5.8.3 Tx Descriptors

A Tx descriptor has the following format:

The descriptor is a two word command.

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	Operation
BufSys Select		Buffer offset (16-bit aligned)														Read/Write

D31	D30	D29	D28	D27	D26	D25	D24	D23	D22	D21	D20	D19	D18	D17	D16	
Descriptor type					Telegram offset (16-bit aligned)											Read/Write

Table 8.101 Tx Descriptor Format

Name	Bits	Size	Description
Buffer offset	1-13	13	16-bit aligned offset of TxBuffer start address in Tx RAM
BufSys Select	14-15	2	Selected buffer system 0: Buffer system A 1: Buffer system B 2: reserved 3: reserved
Telegram offset	17-26	10	16-bit aligned offset after Sercos III CRC in telegram
Descriptor type	28-31	4	See table below

Descriptor types:

Table 8.102 Tx Descriptor Types

Descriptor Type	Descriptor	
0000b	SVDSP	Service data start position
0001b	SVDEP	Service data end position
0010b	RTDSP	Real-time data start position
0011b	RTDEP	Real-time data end position
0100b	FCSP	FCS position, end of transmission
1000b	Port relative data start position, take data out of port dependent buffer	
1001b	Port relative data end position	

Descriptor types 0000b to 0011b write identical data to port 1 and port 2. With descriptors 1000b and 1001b, independent data can be transferred.

8.5.8.4 Tx RAM Pointer

The Tx RAM Pointers are used to define the base addresses for the transmit data buffers. The table has up to 11 entries.

Table 8.103 Tx RAM-Pointer Register Map

Address	Register Symbol	Register Name
4402 01C0h +4h × n	TXBUF[n]_A (n = 0..3)	Transmit buffer [n] base address for buffer system A
4402 01D0h +4h × n	TXBUF[n]_B (n = 0..3)	Transmit buffer [n] base address for buffer system B
4402 01F0h	TXBUF_P1	Transmit buffer base address for port 1 only
4402 01F4h	TXBUF_P2	Transmit buffer base address for port 2 only
4402 01FCh	TXBUF_SVC	Transmit buffer base address for service channel data

8.5.8.5 Overview of Tx Descriptor Engine

Figure 8.16 Tx Descriptor Engine

8.5.8.6 Tx MAC Processing Description

The Tx descriptor processing is started with the detection of a valid ATx.

Depending on the descriptor and the buffer management, the real-time-data-pointer or the service channel data pointer (buffer base addresses) are loaded from the pointer registers.

The data is continuously taken out of the service or the real-time data buffer (in linear ascending order) and inserted at the required position in the respective AT.

After the last commanded AT descriptor, the Tx MAC switches into idle state and waits for the next ATx.

8.5.8.7 Tx MAC Quadbuffering

The active Tx buffer is selected by the Host CPU. The active buffer is sampled with a start descriptor, therefore switching has no effect during transmission of a data descriptor pair. Data consistency must be ensured by firmware while using less than triple buffering.

Table 8.104 Tx Buffer Control & Status Register

Address	Register Symbol	Register Name
4402 020Ch	TXBUFCSR_A	Transmit buffer control for buffer system A
4402 021Ch	TXBUFCSR_B	Transmit buffer control for buffer system B

(1) Tx Buffer Control & Status Register (TXBUFCSR_A/B)

Quadbuffering of the Tx unit is controlled by register “Tx Buffer Control & Status Register” via bit 31 (TxRequest).

8.5.9 Rx MAC Operation

The slave Rx MAC is able to copy data out of the AT or MDT Ethernet telegrams data stream and store this in receive rams (Rx RAM).

8.5.9.1 Rx Descriptors

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	Operation
BufSys Select		Buffer offset (16-bit aligned)														Read/Write

D31	D30	D29	D28	D27	D26	D25	D24	D23	D22	D21	D20	D19	D18	D17	D16	
Descriptor type					Telegram offset (16-bit aligned)											Read/Write

Table 8.105 Rx Descriptor Format

Name	Bits	Size	Description
Buffer offset	1-13	13	16-bit aligned offset of Rx Buffer start address in Rx RAM
BufSys Select	14-15	2	Selected buffer system 0: Buffer system A 1: Buffer system B 2: reserved 3: reserved
Telegram offset	17-26	10	16-bit aligned offset after Sercos III CRC in telegram
Descriptor type	28-31	4	See table below

Descriptor types:

Table 8.106 Rx Descriptor Types

Descriptor Type	Descriptor	
0000b	SVDSP	Service data start position
0001b	SVDEP	Service data end position
0010b	RTDSP	Real-time data start position
0011b	RTDEP	Real-time data end position
0100b	FCSP	FCS position, end of transmission
1000b	(Not used)	
1001b	(Not used)	

8.5.9.2 Rx RAM Pointer

The Rx RAM Pointers are used to define the base addresses for the receive data buffers. The table has up to 14 entries, seven for port 1 and seven for port 2.

Table 8.107 Rx RAM-Pointer Register Map

Address	Register Symbol	Register Name
4402 0180h +4h × n	RXBUF[n]_P1A (n = 0..2)	Receive buffer [n] base address for port 1 and buffer system A
4402 0190h +4h × n	RXBUF[n]_P1B (n = 0..2)	Receive buffer [n] base address for port 1 and buffer system B
4402 019Ch	RXBUF_P1SVC	Receive buffer base address for service channel data at port 1
4402 01A0h +4h × n	RXBUF[n]_P2A (n = 0..2)	Receive buffer [n] base address for port 2 and buffer system A
4402 01B0h +4h × n	RXBUF[n]_P2B (n = 0..2)	Receive buffer [n] base address for port 2 and buffer system B
4402 01BCh	RXBUF_P2SVC	Receive buffer base address for service channel data at port 2

8.5.9.3 Rx MAC Triple Buffering

The active Rx buffer is selected by the Host CPU. The active buffer is sampled at telegram start, therefore switching has no effect during transmission.

Table 8.108 Rx MAC Triple Buffering Register Map

Address	Register Symbol	Register Name
4402 0200h	RXBUFCSR_A	Receive buffer control buffer system A
4402 0204h	RXBUFTV_A	Rx Buffer Telegram Valid A
4402 0208h	RXBUFTR_A	Rx Buffer Telegram Requirements A
4402 0210h	RXBUFCSR_B	Receive buffer control buffer system B
4402 0214h	RXBUFTV_B	Rx Buffer Telegram Valid B
4402 0218h	RXBUFTR_B	Rx Buffer Telegram Requirements B

(1) Rx Buffer Control & Status Register (RXBUFCSR_A/B)

Triplebuffering of the Rx unit is controlled by register “Rx Buffer Control Status Register” via bit 31 (RxRequest).

(2) Rx Buffer Telegram Valid Register (RXBUFTV_A/B)

The valid bits of the telegram status bits are copied to the Rx Buffer Telegram Valid Register to detect invalid data inside actual system buffers.

(3) Rx Buffer Telegram Requirements (RXBUFTR_A/B)

Another way to ensure consistency of current buffers is to prevent buffer change when not all required telegrams are valid. If none is selected, buffer will always change.

8.5.10 Slave Service Channel Operation

The service channel operation is supported by a service channel processor which is able to handle up to eight service channels for a slave device. In this context, the service channel processor handles the following tasks:

- Handshake and Busy Handling
- Receive data buffering up to 254 bytes (depth of write buffer one word)
- Stand alone transmission of up to 254 bytes (depth of read buffer one word plus overhang)

The service channel processor handles the service channel data received and stored in the Rx SVC RAM and transmits information to the master via the Tx SVC RAM. Resources of the Host CPU are low required.

Table 8.109 Slave Service-Channel Operation Register Map

Address	Register Symbol	Register Name
4402 0050h	SVCCSR	SVC Control / Status

8.5.10.1 SVC Control & Status Register (SVCCSR)

Central control of service channels is done by the service channel control/status register SVCCSR. It is provided to define:

- a general enable for all channels
- the port that starts the service channel operation
- the last MDT, after that the service channel processor starts operation
- a manual start one cycle of the service channel processor (for debug purposes)
- a process error flag

CAUTION

If another MDT than 0 is selected to trigger the SVC machine, the MDT valid bits of the preceding MDT telegrams are also used to qualify the trigger. If SVC start mode is active, trigger is done via port timers. Care must be taken to not clear the MDT valid bits until the SVC machine is started. If the bits are cleared before the trigger, the machine will not start and therefore will not process the service containers.

8.5.10.2 Service Container

The eight service container pointers are located at the beginning of the SVC RAM. They contain the offset address of the service containers in the SVC RAM (bit 0 .. 14) and an enable bit for activating the service container.

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	Operation
CEn	MDT Check	Service Container Pointer [n]												MDT Select		Read/Write

Table 8.110 Service Container Pointers Format

Name	Bits	Size	Description
MDT Select	0-1	2	If MDT check is set, the lowest two bits of the container pointer are used to select the according MDT of this container.
Service Container Pointer [n]	(1) 2-13	(13) 12	15-bit pointer, addressing the position of the corresponding service container structure
MDT Check	14	1	If this bit is set, the lower two bits of the container pointer are used for MDT Selection (32-bit alignment of container). Otherwise these bits belong to the container pointer.
CEn	15	1	Service container enable, the container is enabled by setting this bit to ONE.

8.5.10.3 Service Container Pointers

Table 8.111 Service Container RAM Area (relative addresses)

Container Address	Name	Description	
+ 0	SVCRxP	Pointer to service-channel data in Rx RAM, points on service-channel status word.	
+ 2	SVCTxP	Pointer to service-channel data in Tx RAM, points on service-channel control word.	
+ 4	SVCC0	Service-channel control word 0	
+ 6	SVCC1	Service-channel control word 1	
+ 8	SVCC2	Service-channel control word 2	
		End pointer Write-buffer (8 Bit)	Start pointer Write-buffer (8 Bit)
+ A	SVCC3	Service-channel control word 3	
		End pointer Read-buffer (8 Bit)	Start pointer Read-buffer (8 Bit)
+ C	SVCC4	Service-channel control word 4	
+ E	SVCWRB	Write buffer	
—	SVCRDB	Read buffer	
—	SVCRDBO	Read buffer overhang	

Example for service container setup:

- service container pointer 8040h

  container address 40h, service channel enabled
- write buffer 16 words WRDATPT = 0h WRDATLAST = 0Fh
- read buffer 16 words RDDATPT = 10h RDDATLAST = 1Fh

Figure 8.17 Example for Service Container

8.5.10.4 Service Container Control Words

Control words SVCRxP, SVCTxP, SVCC0 to SVCC4 imply all needed information for control of one service channel.

Pointer SVCRxP and SVCTxP define input and output of the service channel. So SVCRxP points to the six slave service channel receive bytes in the Rx RAM and SVCTxP defines the address of the six slave service channel transmit bytes in the Tx RAM.

SVCC0 to SVCC4 contains following information:

Table 8.112 Service Container Control Words

Index	Bit	Name	Function
SVCC0	0	HS_AT	Handshake bit in AT
	1	BUSY_AT	Busy bit in AT, also waiting for host CPU interaction
	2	ERR_AT	Error bit in AT
	3	PROCESS	Process bit in AT
	4-6	ELEM	Data element of present transmission
	7	L/S	Read (0)/write (1) of present transmission
	8-11	—	(Not used)
	12	INT_ELEM_CHANGE	Interrupt – master has modified data element or read/write
	13	INT_END_WRBUF	Interrupt – end of write buffer is reached
	14	INT_END_RDBUF	Interrupt – end of read buffer is reached
	15	INT_END_MDT	Interrupt – master reports end via END_MDT-bit
SVCC1	0	HS_MDT	Handshake bit in MDT
	1	L/S_MDT	Read/write in MDT
	2	END_MDT	End bit in MDT
	3-5	ELEM_MDT	Data element in MDT
	6-15	—	(Not used)
SVCC2	0-7	WRDATPT	Pointer to present position in write buffer
	8-15	WRDATLAST	Pointer to last position in write buffer
SVCC3	0-7	RDDATPT	Pointer to present position in read buffer
	8-15	RDDATLAST	Pointer to last position in read buffer
SVCC4	0-8	—	(Not used)
	9	INT_SC_ERR	Interrupt due to protocol error
	10-15	—	(Not used)

8.5.10.5 Service Processor Operation Details

The service processor principle operation is divided into six steps.

- In the WAIT state, the Busy Bit in SVCC0 is Zero, the service processor waits for a request of the Sercos III master via SVC control.
- On reception of a request of a Sercos III master, the SVC info is transferred to the read buffer, the busy bit is set to 1 and an interrupt INT_END_RDBUF, INT_END_MDT or INT_ELEM_CHANGE is initiated.
- The service processor waits – while the busy bit is set – for an interaction of the Host CPU.
- The Host CPU interprets the request of the master, setup the read or write buffer, deletes the existing interrupt flags and sets the semaphore busy bit to 0.
- With this, the responsibility switches again to the service processor which processes the tasks until either
  - the end of the read buffer is reached – INT_END_RDBUF
  - the end of the write buffer is reached – INT_END_WRBUF
  - the end of the transmission is signaled by the master – INT_END_MDT
  - the element is changed – INT_ELEM_CHANGE
  - or there is an error in the service channel – INT_SC_ERR
- All events result in setting the busy bit and an interaction of the Host CPU.

Flow of slave service channel process:

Figure 8.18 Flow of Slave Service-Channel Process

8.5.10.6 SVC Telegram Data

(1) Format of the Data in the SVC Field of the MDT

Table 8.113 SVC Telegram Data Format

SVC Control (2 Bytes)	SVC Data (4 Bytes)

The Control Field has the following information:

Table 8.114 SVC Telegram Control Field

Bit	Name	Function
0	MHS	Master Handshake
1	R/W	Read / Write Data
2	Last transmission	Transmission in progress (0) Last transmission (1)
3-5	Data block element	Element to transfer
6-15	reserved	—

(2) Format of the Data in the SVC Field of the AT

Table 8.115 SVC Telegram Data Control Field

SVC Status (2 Bytes)	SVC Data (4 Bytes)

The Status Field has the following information:

Table 8.116 SVC Telegram Status Field

Bit	Name	Function
0	AHS	Acknowledge Handshake
1	Busy	Device is busy (1)
2	Error	An error occurred (1)
3	SVC valid	The SVC data is inserted into the telegram (1). This bit is always set by the service channel processor in hardware.
4-15	reserved	—

8.5.10.7 Byte Order of Data in the Telegram

All SVC Data is transmitted in the following order:

Table 8.117 SVC Data Transmitting Order

SVC control (status)	Byte 0 (Bits 0-7)
SVC control (status)	Byte 1 (Bits 8-15)
SVC data	Byte 0 (Bits 0-7)
SVC data	Byte 1 (Bits 8-15)
SVC data	Byte 2 (Bits 16-23)
SVC data	Byte 3 (Bits 24-31)

8.5.11 Hardware Watchdog

After each reset, the watch dog turns inactive (WDCSR: 0x000088CD) and actual count as well as reset count are reset to zero (WDCNT: 0x00000000).

The first time, when the magic pattern is written to the lower 16 bits of WDCSR (trigger signal), the watchdog becomes active. It's is deactivated by writing inverse magic pattern (0x00007732).

Each trigger copies the reset count value to actual count in WDCNT. During cyclic communication, the logic detects the port, where the first MST arrives. Receiving a MST at this port, counts down the actual count as long as actual count is greater zero.

CAUTION

The watchdog signals alarm and disables data introduction into frames when it is active and actual count equals zero. Alarm and data introduction can only be recovered with deactivation (writing inverse magic pattern).

Table 8.118 Hardware Watchdog Register Map

Address	Register Symbol	Register Name
4402 0068h	WDCSR	Watchdog Control & Status
4402 006Ch	WDCNT	Watchdog Counter

8.5.12 MST Monitoring

Table 8.119 MST Monitoring Register Map

Address	Register Symbol	Register Name
4402 00E0h	MSTLMAX	Maximum sequential MST losses
4402 00E4h	MSTLSUM	Sum MST losses
4402 00E8h	MSTLACT	Actual sequential MST losses

(1) Maximum Sequential MST Losses (MSTLMAX)

Maximum count for sequential MST losses at port 1 and port 2. Note that MST's have to be lost at both ports for each count. An MST loss is detected by the overrun of TCNT0.

When the internal counter reaches half of this value (round up), this module will set interrupt status bit 10. Interrupt status bit 11 will be set, when current sequential losses equals this maximum.

(2) Sum MST Losses (MSTLSUM)

Counter for MST losses at port 1 and port 2. Note that both MST's have to be lost for a count.

Each write access on this counter resets it to zero.

Does not increment anymore if counter has reached 0xFFFF.

(3) Actual Sequential MST Losses (MSTLACT)

This register shows the actual amount of sequential MST losses.

8.5.13 IP Channel Operation

The Sercos III IP telegram processing

- monitors the received data stream to detect the frame type and starts operation when an Ethernet frame type is detected,
- handles the data transfers in and out of the SRAM
- stores Ethernet telegrams in special collision buffers, when Tx path is occupied by internal Tx MAC,
- sends telegrams out the collision buffer preferred to Tx MAC.

Table 8.120 IP Channel Operation Register Map

Address	Register Symbol	Register Name
4402 0080h	MAC1P1_0	MAC Address 0
4402 0084h	MAC1P1_1	MAC Address 1
4402 0090h	IPCSR1	IP Status-/Control register port 1
4402 0094h	IPCSR2	IP Status-/Control register port 2

8.5.13.1 MAC Address Register

The MAC address, a six byte value, is used for comparison with the destination address of IP telegrams.

8.5.13.2 IP Control and Status Register (IPCSR 1/2)

Behavior of status bits:

Status bits could not be cleared by writing ZERO into the status register.

Tx status bits:

- IPTxBufRdy is cleared when Tx buffer is full.
- IPTxBufEmpty is cleared when at least one descriptor is on the Tx stack register.

Rx status bits:

- IPRxRdy is set when at least one descriptor is on the Rx stack register.
- IPRxBufFull is set when all descriptors on the Rx stack register are used.

8.5.13.3 Collision Buffer Operation

The device uses two 2 Kbyte collision buffers between port1 & port2 and port2 & port1 to buffer forwarding standard Ethernet frames, when the internal Tx MAC allocates the Tx line.

The device uses the “cut through” mechanism, this means:

- an Ethernet frame is buffered until the destination address is received
- if the destination address matches the internal MAC address, the frame will not be forwarded
- Otherwise, if the desired Tx port isn’t allocated by the internal Tx MAC and the remaining IP Slot is able to carry the required frame length, the Ethernet frame is forward to this Tx port
- Otherwise, Ethernet frame is buffered in the collision buffer RAM until the next IP channel gets active

The Tx MAC has lower priority than the collision buffer path. This means, when a new IP channel gets active and a frame is still buffered in the collision buffer RAM, the transmission of collision buffer content has highest priority.

8.5.13.4 Transmitting IP Telegrams

Table 8.121 Transmitting IP Telegram Register Map

Address	Register Symbol	Register Name
4402 00A8h	IPTXS1	IP Transmit Stack port 1
4402 00ACh	IPTXS2	IP Transmit Stack port 2

(1) Operation Description

All transmit instructions of the host are stored in the transmit stacks. The device has a transmit stack for both ports, register address A8h and ACh.

When IP telegram should be transmitted the host loads the data, the complete Ethernet frame without preamble and FCS into the Tx RAM area and loads the related TxIP descriptor onto the corresponding transmit stack, register IPTXS.

Table 8.122 Ethernet Frame Structure in Tx RAM

DA	SA	Type	Data

case 1:

- the IPTxEN bit is set and the MAC is enabled

case 2:

- the IPTxEN bit isn’t active (SIII mode)
- the MAC is enabled
- the Tx path isn’t used
- the IP channel time slot is active
- remaining transmission time to IP channel close is sufficient

(2) IP Transmit Stack Port 1/2 (IPTXS1/2)

To transmit an Ethernet frame the TxIP descriptor has to be written into the transmit stack register.

IPTXS1 for port1 and/or IPTXS2 for port 2. Writing IPTXS1 (0A8h) sends a frame to port 1. As long as “Count” is not zero, reading IPTXS1/2 shows the values of the actual frame in progress. Otherwise the currently written data can be read. If no new data was written, the readback values are not valid.

(3) Definition of TxBuffer

The Tx buffers are defined by the TxIP descriptors itself, so no more other definition registers are needed.

Example)

Host wants to send identical Ethernet telegrams on two ports

- so the host sets identical lengths and start segments (10h) in IPTXS1 and IPTXS2

or host wants to send different Ethernet telegrams on two ports

- so the host sets different segments in IPTXS1 (10h) and IPTXS2 (18h)

Figure 8.19 Example for Tx-buffers

8.5.13.5 Receiving IP Frames

The Rx MAC can receive up to 8 frames on each port. The number of frames is also limited by the memory that is available for NRT communication.

Table 8.123 Receiving IP Frames Register Map

Address	Register Symbol	Register Name
4402 0098h	IPRRS1	IP Rx RAM Segment port 1
4402 009Ch	IPRRS2	IP Rx RAM Segment port 2
4402 00A0h	IPRXS1	IP Receive Stack port 1
4402 00A4h	IPRXS2	IP Receive Stack port 2

(1) IP Rx RAM Segment Port 1/2 (IPRRS1/2)

The receive buffers for standard Ethernet frames in the Rx RAM are defined by the segment registers IPRRS1 and IPRRS2, whereby IPRRS1 is assigned to port 1 and IPRRS2 is assigned to port 2. The internal representation of this register is only updated if IPRxEn of appropriate IPCSR is inactive.

Example)

Host reserves 4.5K receive buffer for port 1 and port 2. Remember receive segments have a size of 256 bytes. So host sets for each segment register (IPRRS) 17 (11h) segments (2.25K) as follows

- first receive buffer segment register to 21h – 10h (last segment – first segment) and
- second receive buffer segment register to 2Fh – 24h (last segment – first segment).

Figure 8.20 Example for Receiving IP Frames

(2) IP Receive Stack Port 1/2 (IPRXS1/2)

If an Ethernet frame is received correctly the RxIP descriptor is put onto the receive stack IPRXS1 for port 1 and/or IPRXS2 for port 2

CAUTION

The actual top descriptor on the receive stack is cleared by write to the Rx stack register by the host.

(3) Rx MAC Processing

The processing of the Rx MAC operates on port 1 and port 2.

Rx MAC Operation:

- The Rx MAC's will accept standard Ethernet frames at any time, when the
  - destination address matches the MAC address, or
    a broadcast is addressed (broadcasts must be enabled)
  - Rx buffer isn't full; otherwise the error counter aDiscardResRxBuf is incremented
- When the complete frame is received without error, the descriptor is stored onto the stack and the reception is indicated by setting the IPRxRdy and, if the Rx buffer is full, the IPRxBufFull bit is set. An IPRxEvent (IPRxRdy or IPRxBufFull) is generated and, when the corresponding interrupt is enabled, an interrupt is requested to the host.
- If the frame is received with a FCS error, the frame is discarded and the counter aFCSErrors is incremented.
- If the wrong aligned frame is received, the frame is discarded and the counter aAlignmentErrors is incremented.
- If the IPRxEN bit is inactive (SIII Mode) and the Rx MAC is receiving a standard Ethernet frame violating the IP channel boundaries, "Event_IPChannel_Open" and "Event_IPChannel_RxClose", the error counter aIPChannelViolation is incremented.

Frame structure in Rx RAM:

Each received standard Ethernet frame is stored in the following manner.

Table 8.124 Frame Structure in Rx RAM

DA	SA	Type	Data	FCS

8.5.13.6 Remaining Frame Length (IPLASTFL)

In order to use the maximum bandwidth of the IP channel, the application may set the remaining bytes for IP frames at the last transmit event of the port timers. If the length of a frame inside the queue is known (store and forward), it can be sent also after the last transmit event.

The effect of IPLASTFL can be seen in following figure. If IPLASTFL is not used, packet 3 as well as packets b, c and d would be shifted to the following cycle with delay and a lot of bandwidth wasted. Furthermore there shall be additional space left for at least one IFG to the IP Tx closing event.

Table 8.125 Remaining frame length Register Map

Address	Register Symbol	Register Name
4402 00B0h	IPLASTFL	Remaining frame length

Figure 8.21 Effect of IPLASTFL

8.5.13.7 Frame and Error Counters

Eight counters per port are implemented to sum valid received and transmitted frames as well as five different communication errors and an additional Sercos III sum error counter. The error counters saturate at 0xFFFF. All counters can be reset by writing to aSercosErrorCount.

Table 8.126 Frame and Error Counters Register Map

Address	Register Symbol	Register Name
4402 00C0h	IPFRXOK	IP aFramesReceivedOK
4402 00C4h	IPFTXOK	IP aFramesTransmittedOK
4402 00C8h	IPFCSERR	IP aFCSErrors
4402 00CCh	IPALGNERR	IP aAlignmentErrors
4402 00D0h	IPDISRXB	IP aDiscardResRxBuf
4402 00D4h	IPDISCLB	IP aDiscardResColBuf
4402 00D8h	IPCHVIOL	IP aIPChannelViolation
4402 00DCh	aSercosErrorCount	Sercos III Error Counter

(1) IP aFramesReceivedOK

Counts all received frames without error on the port (includes forwarded and discarded frames due low resource).

(2) IP aFramesTransmittedOK

Counts all transmitted frames on the port (includes forwarded frames).

(3) IP aFCSErrors

Counts the received Ethernet frames with defective frame check sequence FCS or RxER indications.

(4) IP aAlignmentErrors

The counters increment when a defective ethernet frame is detected. Defective frames are wrong aligned frames.

CAUTION

Frames where the preamble and SFD (start frame delimiter) isn't detected will be discarded.

(5) IP aDiscardResRxBuf

The counters counts the discarded receive Ethernet frames in case of missing Rx buffer resources.

(6) IP aDiscardResColBuf

The counters count the discarded forwarding Ethernet frames in case of missing collision buffer resources.

(7) IP aIPChannelViolation

The counters increments on Ethernet frames that violate IP channel time boundaries.

(8) Sercos III Error Counter

The counter counts Sercos III Ethernet frames that have a wrong FCS or are misaligned. This counter saturates at 0xFFFF and can be cleared by writing to it.

8.5.14 MDIO Control

This register can be used to read and write PHY registers for configuration and diagnosis.

It is possible to use just one interface and select the PHY's with a unique address. The addressing of the PHY's is implementation dependent.

Table 8.127 MDIO Control Register Map

Address	Register Symbol	Register Name
4402 0074h	MIICSR	MDIO Control / Status Register

8.5.15 Debug Output Control

Debug output control register is used to output internal signals on test signal outputs S3_TESTPIN1 and S3_TESTPIN2.

Table 8.128 Debug Output Control Register Map

Address	Register Symbol	Register Name
4402 0078h	DBGOCR	Debug output control

Selectable output signals and corresponding codes:

Table 8.129 Debug Output Signals and Corresponding Codes

Value	Signal Slave	Description
0	Port 1 MST	MST pulse from the Rx MAC of port 1 (40 ns duration)
1	Port 2 MST	MST pulse from the Rx MAC of port 2 (40 ns duration)
2	TMST	TMST signal after MST generator
3	CON_CLK	CON_CLK from the TCNT timer
4	DIV_CLK	DIV_CLK from the DIV_CLK unit (only if present)
5	TCNT Reload	Overflow of the TCNT timer
6	Port 1 TCNT Reload	Overflow of the Port 1 timer
7	Port 2 TCNT Reload	Overflow of the Port 2 timer
8	Port 1 IP Open	Port 1 IP window
9	Port 1 IP Open Write	Port 1 IP transmit window
10	Port 2 IP Open	Port 2 IP window
11	Port 2 IP Open Write	Port 2 IP transmit window
12	Port 1 MST Window Open	Port 1 MST window
13	Port 2 MST Window Open	Port 2 MST window
14	Port 1 Rx Frame	Reception of a frame on port 1
15	Port 2 Rx Frame	Reception of a frame on port 2

8.5.16 Sercos III LED Control

Table 8.130 Sercos III LED Control Register Map

Address	Register Symbol	Register Name
4402 005Ch	S3LED	Sercos III LED Control

8.5.16.1 S3LED

The Sercos III LED is controlled over two colors. The hardware toggles between both with roughly 4 or 2 Hz.

MSB of each Color[n] bit is output to S3_LED_RD, LSB of each Color[n] bit is output to S3_LED_GN.

Figure 8.22 LED Control

8.5.16.2 Active LED

Table 8.131 Active LED's Behavior

Sercos III Phase	Description
NRT	Flashes with roughly 2 Hz during activity but at least one active state of 250 ms. With active collision buffers, both port LED's flash synchronously, otherwise with 180° phase shift.
CP0-CP3 or CP5	LED's are permanently switched on at active ports as long as there is traffic on the network. At inactive ports LED's flash with roughly 2 Hz during traffic.
CP4	Asynchronous: Same as CP0-CP3 Synchronous: Flashes with roughly 4 Hz during activity, synchronous to other slaves.

Section 9 R-IN Engine Accessory Register

All R-IN Engine Accessory registers can be accessed by 32-bit units.

9.1 Register Map

Table 9.1 R-IN Engine Accessory Register List

Address	Register Symbol	Register Name
400F 2004h	IDCODE	IDCODE register
400F 2010h + 4h × n	SCRATCH[n] (n = 0..3)	Scratch register [n]
400F 2100h	RINSPCMD	R-IN System Protect Command register
400F 2110h	RTOSRST	HW-RTOS and HW-RTOS GMAC reset register

9.2 Register Description

9.2.1 IDCODE — IDCODE Register

Address: 400F 2004h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	IDCODE															
Value after reset	0	1	0	1	0	0	1	0	0	1	0	0	1	0	0	1

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	IDCODE															
Value after reset	0	1	0	0	1	1	1	0	0	0	1	1	0	0	1	1

Table 9.2 IDCODE Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	IDCODE	This register is used to identify the R-IN Engine. When this register is read, RIN3 is returned in ASCII code.	R

9.2.2 SCRATCH[n] — Scratch register [n] (n = 0..3)

Address: 400F 2010h + 4h × n

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	SCRATCH[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SCRATCH[n]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 9.3 SCRATCH[n] Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	SCRATCH[n]	General register[n]	R/W

9.2.3 RINSPCMD — R-IN System Protect Command Register

Address: 400F 2100h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RINSP CMD
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 9.4 RINSPCMD Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b1	Reserved	Set to zero on Write. ignore on Read.	R
b0	RINSPCMD	Permit Write Access to Protected Register 0: Prohibit write access. 1: Permit write access.	R/W

9.2.4 RTOSRST — HW-RTOS and HW-RTOS GMAC Reset Register

Address: 400F 2110h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	OSRST
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 9.5 RTOSRST Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b1	Reserved	Set to zero on Write. ignore on Read.	R
b0	OSRST	HW-RTOS and HW-RTOS GMAC reset control 0: Assert reset. 1: De-assert reset.	R/W

9.3 Usage Notes

9.3.1 R-IN Engine Accessory Register

9.3.1.1 R-IN System Protect Command Register (RINSPCMD)

Protection release sequence:

1. Write 0000_00A5h to RINSPCMD register
2. Write 0000_0001h to RINSPCMD register
3. Write 0000_FFFEh to RINSPCMD register
4. Write 0000_0001h to RINSPCMD register

Protection sequence:

1. Write 0b to LSB of RINSPCMD register. (Write protect enabled.)

Figure 9.1 R-IN System Protect Command Register Operation Sequence

9.3.1.2 HW-RTOS Reset Register (RTOSRST)

The RTOSRST register is used to control the reset status of HW-RTOS GMAC by software. The reset value of RTOSRST is 0, this means that HW-RTOS GMAC is still in the reset state unless RTOSRST is set.

To reset HW-RTOS GMAC during operation, write 0 to RTOSRST. Then, write 1 to release the reset state after confirming that the bit is set to 0.

This register is write-protected by the R-IN system protect command register (RINSPCMD). To perform a write to this register, use the RINSPCMD register to release write-protection.

Section 10 Ethernet Accessory Register

All Ethernet Accessory registers can be accessed by 32-bit units.

10.1 Register Map

Table 10.1 Ethernet Accessory Register List

Address	Register Symbol	Register Name
4403 0000h	PRCMD	Ethernet Protect Register
4403 0004h	ESIDCODE	Ethernet Peripherals IDCODE Register
4403 0008h	MODCTRL	Mode Control Register
4403 000Ch	PTPMCTRL	PTP Mode Control Register
4403 0014h	PHYLNK	Ethernet PHY Link Mode Register
4403 0020h	PTCTRL	PortTrigger Control Register
4403 0024h	DMACTRL	DMAC Control Register
4403 0100h + 4h × (m − 1)	CONVCTRL[m] (m = 1 to 5)	RGMII/RMII Converter[m] Control Register
4403 0114h	CONVRST	RGMII/RMII Converter Reset Control Register
4403 0200h	ECATOFFADR	EtherCAT PHY Offset Address Register
4403 0204h	ECATOPMOD	EtherCAT Operation Mode Register
4403 0208h	ECATDBGC	EtherCAT Debug Control Register
4403 0280h	SCINTCON	SERCOS Interrupt Control Register
4403 0304h	SWCTRL	A5PSW Control Register
4403 0308h	SWDUPC	A5PSW Duplex Mode Register
4403 0500h	RMTAGCTRL	HW-RTOS GMAC Management TAG Control Register
4403 0600h	HSRMOD	HSR Mode Indication Register

10.2 Register Description

10.2.1 PRCMD — Ethernet Protect Register

Address: 4403 0000h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PRCMD
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 10.2 PRCMD Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b1	Reserved	Reserved	R
b0	PRCMD	Permit write Access to Protected Registers 0: Prohibit write access. 1: Permit write access	R/W

10.2.2 ESIDCODE — Ethernet Peripherals IDCODE Register

Address: 4403 0004h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	ES_IDCODE															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ES_IDCODE															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	X	X

Table 10.3 ESIDCODE Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b0	ES_IDCODE	ID Code (RZ/N1D): 0x0000_0001 (RZ/N1S, RZ/N1L): 0x0000_0002	R

10.2.3 MODCTRL — Mode Control Register

Address: 4403 0008h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	SW_MODE				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 10.4 MODCTRL Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b5	Reserved	Reserved	R
b4 to b0	SW_MODE	Select Ethernet interface connection. Please refer to *Section 8.2.1, Internal Connection of Ethernet Ports*, in the *RZ/N1D Group, RZ/N1S Group, RZ/N1L Group User's Manual: System Introduction, Multiplexing, Electrical and Mechanical Information*.	R/W

10.2.4 PTPMCTRL — PTP Mode Control Register

Address: 4403 000Ch

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	RGMII_CLKSEL	PTP_MODE			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 10.5 PTPMCTRL Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b5	Reserved	Reserved	R
b4	RGMII_CLKSEL	Select clock source of RGMII 0: 125 MHz (generated from PLL) 1: RGMII_REFCLK (125 MHz from external pin) ***Note)*** This bit is ignored when CONV_MODE is not in RGMII modes.	R/W
b3 to b0	PTP_MODE	Select clock source of PTP timer for GMAC1 and GMAC2 0000b: Stop (Low Level) 0001b: RGMII_REFCLK (125 MHz from external pin) 0010b: 125 MHz (generated from PLL) 0011b: 50 MHz (generated from PLL) 0100b: 25 MHz (generated from PLL) Others: Reserved. (Do not set.)	R/W

10.2.5 PHYLNK — Ethernet PHY Link Mode Register

Address: 4403 0014h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	S3LNK		—	CATLNK			SWLNK			
Value after reset	0	0	0	0	0	0	1	1	0	1	1	1	0	0	0	0

Table 10.6 PHYLNK Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b10	Reserved	Reserved	R
b9, b8	S3LNK	Select SERCOS3 Link Signal Active Level 0: Low active 1: High active (Default) ***Note*)** Each bit corresponds to link signal below. S3LNK[0] control S3_MII_LINKP[1] S3LNK[1] control S3_MII_LINKP[2]	R/W
b7	Reserved	Reserved	R
b6 to b4	CATLNK	Select EtherCAT Link Signal Active Level 0: Low active 1: High active (Default) ***Note*)** Each bit corresponds to link signal below. CATLNK[0] control CAT_MII_LINK[0] CATLNK[1] control CAT_MII_LINK[1] CATLNK[2] control CAT_MII_LINK[2]	R/W
b3 to b0	SWLNK	Select A5PSW Link Signal Control Active Level 0: High active (Default) 1: Low active ***Note*)** Each bit corresponds to link signal below. SWLNK[0] control SWITCH_MII_LINK[5] SWLNK[1] control SWITCH_MII_LINK[4] SWLNK[2] control SWITCH_MII_LINK[3] SWLNK[3] control SWITCH_MII_LINK[2]	R/W

10.2.6 PTCTRL — PortTrigger Control Register

Address: 4403 0020h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	TRG_SEL		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 10.7 PTCTRL Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b3	Reserved	Reserved	R
b2 to b0	TRG_SEL	Select Port Trigger Source (GPIO_TRIGGER[3:0]). (Refer to **Section 10.4.2, PortTrigger Control Register**.)	R/W

10.2.7 DMACTRL — DMAC Control Register

Address: 4403 0024h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	DREQ_WCNT								—	DREQ_ERR						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	DREQ_BSY							DREQ_SEL	DREQ_EN						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 10.8 DMACTRL Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b24	DREQ_WCNT	DMA transfer request wait control. 11111111b: 256 clock cycle wait. : 00000000b: 1 clock cycle wait.	R/W
b23	Reserved	Reserved	R
b22 to b16	DREQ_ERR	DMA Error Status (bit0=ETH_DMA_Request[0] .. bit6= ETH_DMA_Request[6]) 0: Normal status 1: Error status	R/W
b15	Reserved	Reserved	R
b14 to b8	DREQ_BSY	DMA busy Status (bit0=ETH_DMA_Request[0] .. bit6= ETH_DMA_Request[6]) 0: No operation 1: DMA transferring	R
b7	DREQ_SEL	DMA Request Select (For ETH_DMA_Request[0] and ETH_DMA_Request[1]) 0: EtherCAT 1: Sercos3	R/W
b6 to b0	DREQ_EN	Enable DMA Request (bit0=ETH_DMA_Request[0] .. bit6= ETH_DMA_Request[6]) 0: Disable DMA request 1: Enable DMA request	R/W

10.2.8 CONVCTRL[m] — RGMII/RMII Converter[m] Control Register (m = 1 to 5)

Address: 4403 0100h + 4h × (m − 1)

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	RGMII_DLY_TYPE		RGMII_TX_TYPE		—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RGMII_SPEED		RGMII_DUPLEX	RGMII_LINK	—	RMII_CRS_MODE	RMII_RX_ER_EN	FULLD	—	—	—	CONV_MODE				
Value after reset	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0

Table 10.9 CONVCTRL[m] Register Contents (1/2)

Bit Position	Bit Name	Function	R/W
b31 to b28	Reserved	Reserved	R
b27, b26	RGMII_DLY_TYPE	Indicate RGMII clock delay control type. 00b: Not applicable. Others: Reserved	R
b25, b24	RGMII_TX_TYPE	Indicate RGMII TX clock control type. 01b: Use normal clock input. Others: Reserved	R
b23 to b16	Reserved	Reserved	R
b15, b14	RGMII_SPEED	Indicates Link speed. (Only for RGMII mode*1) 00b: 2.5 MHz 01b: 25 MHz 10b: 125 MHz 11b: Reserved	R
b13	RGMII_DUPLEX	Indicates duplex status. (Only for RGMII mode*1) 0: Half duplex 1: Full duplex	R
b12	RGMII_LINK	Indicates link status. (Only for RGMII mode*1) 0: Down 1: Up	R
b11	Reserved	Reserved	R
b10	RMII_CRS_MODE	CRS detection mode. (Only for RMII mode) 0: CRS or TX_EN 1: CRS or TX_EN or RX_DV	R/W
b9	RMII_RX_ER_EN	GMII[m]_RXER pin input control. (Only for RMII mode) 0: Disable (always 0) 1: Enable	R/W
b8	FULLD	Duplex Mode (For RMII/RGMII mode.) 0: Half duplex 1: Full duplex	R/W
b7 to b5	Reserved	Reserved	R

Table 10.9 CONVCTRL[m] Register Contents (2/2)

Bit Position	Bit Name	Function	R/W
b4 to b0	CONV_MODE	Converter Operation Mode 00000b and 10000b: MII mode (Through mode) 00100b: RMII mode 10 Mbps REF_CLK input 00101b: RMII mode 100 Mbps REF_CLK input 10100b: RMII mode 10 Mbps REF_CLK output 10101b: RMII mode 100 Mbps REF_CLK output 01000b and 11000b: RGMII mode 10 Mbps 01001b and 11001b: RGMII mode 100 Mbps 01010b and 11010b: RGMII mode 1000 Mbps Others: Reserved (Do not use.)	R/W

Note 1. This status function is only supported when external phy have RGMII optional specification that inter-frame signals are placed onto the RXD[3:0] signals.

10.2.9 CONVRST — RGMII/RMII Converter Reset Control Register

Address: 4403 0114h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	PHYIF_RSTn				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 10.10 CONVRST Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b5	Reserved	Reserved	R
b4 to b0	PHYIF_RSTn	Release reset of RGMII Converter 0: Reset RGMII Converter 1: Activate RGMII Converter	R/W

10.2.10 ECATOFFADR — EtherCAT PHY Offset Address Register

Address: 4403 0200h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	OADD				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 10.11 ECATOFFADR Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b5	Reserved	Reserved	R
b4 to b0	OADD	PHY Address Offset for EtherCAT	R/W

10.2.11 ECATOPMOD — EtherCAT Operation Mode Register

Address: 4403 0204h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	I2CSIZE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 10.12 ECATOPMOD Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b1	Reserved	Reserved	R
b0	I2CSIZE	Select I2C memory size for EtherCAT 0: Up to 16 Kbits EEPROM 1: 32 Kbits to 4 Mbits EEPROM	R/W

10.2.12 ECATDBGC — EtherCAT Debug Control Register

Address: 4403 0208h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	TXSFT2		TXSFT1		TXSFT0	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 10.13 ECATDBGC Register Contents

Bit Position	Bit Name	Function	R/W
b31, b30	Reserved	Always Write 0	R/W
b29 to b6	Reserved	Reserved	R
b5, b4	TXSFT2	Select Tx clock (GMII3_TXCLK) delay of EtherCAT slave controller 00b: 0 ns 01b: 10 ns 10b: 20 ns 11b: 30 ns	R/W
b3, b2	TXSFT1	Select Tx clock (GMII4_TXCLK) delay of EtherCAT slave controller 00b: 0 ns 01b: 10 ns 10b: 20 ns 11b: 30 ns	R/W
b1, b0	TXSFT0	Select Tx clock (GMII5_TXCLK) delay of EtherCAT slave controller 00b: 0 ns 01b: 10 ns 10b: 20 ns 11b: 30 ns	R/W

10.2.13 SCINTCON — SERCOS Interrupt Control Register

Address: 4403 0280h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	SC_DIV_SEL		—	—	SC_CON_SEL	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 10.14 SCINTCON Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b6	Reserved	Reserved	R
b5, b4	SC_DIV_SEL	S3_DIVCLK Interrupt (SERCOS3_DIVCLK_Int) detection type control 00b: Through mode (Original S3_DIVCLK) 01b: Rising edge trigger 10b: Falling edge trigger 11b: Both edge trigger	R/W
b3, b2	Reserved	Reserved	R
b1, b0	SC_CON_SEL	S3_CONCLK Interrupt (SERCOS3_CONCLK_Int) detection type control 00b: Through mode (Original S3_CONCLK) 01b: Rising edge trigger 10b: Falling edge trigger 11b: Both edge trigger	R/W

10.2.14 SWCTRL — A5PSW Control Register

Address: 4403 0304h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	STRAP_HUB_ENB	STRAP_SX_ENB
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	SET1000				SET10			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

Table 10.15 SWCTRL Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b18	Reserved	Reserved	R
b17	STRAP_HUB_ENB	This bit and STRAP_SX_ENB bit select initial port operation by controlling PORT_ENA and AUTH_PORT[n] register initial value. (set while RSTN_B bit of PWRCTRL_SWITCH is 0) 0: All ports are disabled if STRAP_SX_ENB bit is 0, otherwise all ports except management port are enabled. 1: Port0 and port1 are only enabled if STRAP_SX_ENB bit is 0, otherwise all ports are enabled.	R/W
b16	STRAP_SX_ENB	This bit and STRAP_HUB_ENB bit select initial port operation by controlling PORT_ENA and AUTH_PORT[n] register initial value. (set while RSTN_B bit of PWRCTRL_SWITCH is 0) 0: All ports are disabled if STRAP_HUB_ENB bit is 0, otherwise port0 and port1 are only enabled. 1: All ports except management port are enabled if STRAP_HUB_ENB bit is 0, otherwise all ports are enabled.	R/W
b15 to b8	Reserved	Reserved	R
b7 to b4	SET1000	This bit function is same as ETH_SPEED bit of COMMAND_CONFIG_P[n] register (n=0..3). Port control to select use of 1000 Mbps. Bit0 = port0…Bit3 = port3. 0: Not 1000 Mbps 1: 1000 Mbps ***Note*)** It is recommended that these bits are set to 0. Use ETH_SPEED bit of COMMAND_CONFIG_P[n] register instead.	R/W
b3 to b0	SET10	This bit is no influence except PHYSPEED bit of STATUS_P[n] register (n=0..3). When using the HUB function, set these bits as they are reflected appropriately in STATUS_P[n].PHYSPEED. See STATUS_P[n].PHYSPEED for speed settings.	R/W

10.2.15 SWDUPC — A5PSW Duplex Mode Register

Address: 4403 0308h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	PHY_DUPLEX			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

Table 10.16 SWDUPC Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b4	Reserved	Reserved	R
b3 to b0	PHY_DUPLEX	Configures each port's MAC for full duplex operation or half duplex. Bit0 = port0…Bit3 = port3. This bit is ignored if HD_ENA bit of COMMAND_CONFIG_P[n] register is 0. 0: Half duplex. 1: Full duplex ***Note)*** It is recommended that these bits are set to 0. Use HD_ENA bit of COMMAND_CONFIG_P[n] register instead.	R/W

10.2.16 RMTAGCTRL — HW-RTOS GMAC Management TAG Control Register

Address: 4403 0500h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	MGMT_TAG															
Value after reset	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MGMT_ENB
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 10.17 RMTAGCTRL Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b16	MGMT_TAG	EtherType of management tag	R/W
b15 to b1	Reserved	Reserved	R
b0	MGMT_ENB	Enable to insert management tag to Ethernet frame. 0: Disable 1: Enable ***Note)*** When the insertion of the management tag is enabled by this bit, it is also required to enable the management tag function of the MGMT_TAG_CONFIG register of A5PSW. For details, refer to **Section 3.6.3, Management TAG Control**.	R/W

10.2.17 HSRMOD — HSR Mode Indication Register

Address: 4403 0600h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	HSR_SEL
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 10.18 HSRMOD Register Contents

Bit Position	Bit Name	Function	R/W
b31 to b1	Reserved	Reserved	R
b0	HSR_SEL	Indicate HSR mode. HSR mode is chosen by MODCTRL register. 0: Not HSR mode (Buffer RAM is used for HW-RTOS GMAC) 1: HSR mode (Buffer RAM is used for HSR)	R

10.3 Operation

Please refer to *Section 8.3.2, ETHMODE_SET*, in the *RZ/N1D Group, RZ/N1S Group, RZ/N1L Group User's Manual: System Introduction, Multiplexing, Electrical and Mechanical Information*.

10.4 Usage Notes

10.4.1 Ethernet Protect Register

Address	Register Symbol	Register Name
4403 0000h	PRCMD	Ethernet Protect register

PRCMD register is used to permit write access to protected registers. This register mitigates the risk which application system carries out unusual operation by software runaways.

The list of registers protected by Ethernet Protect Register is shown below.

Table 10.19 The List of Registers Protected by Ethernet Protect Register

Address	Register Symbol	Register Name
4403 0008h	MODCTRL	Mode Control register
4403 000Ch	PTPMCTRL	PTP Mode Control register
4403 0014h	PHYLNK	Ethernet PHY Link Mode register
4403 0020h	PTCTRL	PortTrigger Control register
4403 0024h	DMACTRL	DMAC Control register
4403 0114h	CONVRST	RGMII/RMII Converter Reset Control register
4403 0200h	ECATOFFADR	EtherCAT PHY Offset address register
4403 0204h	ECATOPMOD	EtherCAT Operation Mode register
4403 0208h	ECATDBGC	EtherCAT Debug Control register
4403 0280h	SCINTCON	SERCOS Interrupt Control register
4403 0304h	SWCTRL	A5PSW Control register
4403 0308h	SWDUPC	A5PSW Duplex Mode register
4403 0500h	RMTAGCTRL	HW-RTOS GMAC Management TAG Control register

Before set PRCMD bit to 1, protected registers are refused write access. (No error, but do not updated by write data.)

PRCMD bit set procedure:

1. Write 0000_00A5h to PRCMD register
2. Write 0000_0001h to PRCMD register
3. Write 0000_FFFEh to PRCMD register
4. Write 0000_0001h to PRCMD register

PRCMD bit clear procedure:

1. Write 0b to LSB of PRCMD register. (Write protect enabled.)

Figure 10.1 Ethernet Protect Command Sequence

10.4.2 PortTrigger Control Register

Address	Register Symbol	Register Name
4403 0020h	PTCTRL	PortTrigger Control register

PTCTRL register is used to select Port Trigger Source for GPIO_TRIGGER[3:0]. The selection table is shown below.

Table 10.20 Port Trigger Source Selection

TRG_SEL[2:0]	GPIO_TRIGGER[0]	GPIO_TRIGGER[1]	GPIO_TRIGGER[2]	GPIO_TRIGGER[3]
0	ETHCAT_SYNC_Int[0]	ETHCAT_SYNC_Int[1]	MAC_PPS[0]	MAC_PPS[1]
1	SERCOS3_Int[0]	SERCOS3_Int[1]	MAC_PPS[0]	MAC_PPS[1]
2	ETHCAT_SYNC_Int[0]	ETHCAT_SYNC_Int[1]	MAC_TRIG[1]	MAC_PPS[0]
3	SERCOS3_Int[0]	SERCOS3_Int[1]	MAC_TRIG[1]	MAC_PPS[0]
4	(Reserved)	MAC_TRIG[1]	MAC_PPS[0]	MAC_PPS[1]
5 to 7	(Reserved)	(Reserved)	(Reserved)	(Reserved)

10.4.3 Ethernet PHY Link Mode Register

Address	Register Symbol	Register Name
4403 0014h	PHYLNK	Ethernet PHY Link Mode register

Link signal input control register. Default PHY link signal is active high. If a link LED output from PHY is active low, change setting of this register.

10.4.4 DMAC Control Register

Address	Register Symbol	Register Name
4403 0024h	DMACTRL	DMAC Control Register

This register select and control source of DMA requests (ETH_DMA_Request[n] (n = 0..6)). The selection table by DREQ_SEL bit is shown below.

Table 10.21 DMA Request Source Selection Table

DMA Request Line*1	DREQ_SEL	Source Signal	Function
ETH_DMA_Request[0]	0	CAT_SYNC0	EtherCAT SYNC0
ETH_DMA_Request[0]	1	SERCOS3_Int[0]	SERCOS3 interrupt Port 1 (Port A)
ETH_DMA_Request[1]	0	CAT_SYNC1	EtherCAT SYNC1
ETH_DMA_Request[1]	1	SERCOS3_Int[1]	SERCOS3 interrupt Port 2 (Port B)
ETH_DMA_Request[2]	0 or 1	MAC_PPS[0]	GMAC1 Pulse Per Second output0 (PPS0 of MAC1)
ETH_DMA_Request[3]	0 or 1	MAC_PPS[1]	GMAC1 Pulse Per Second output1 (PPS1 of MAC1)
ETH_DMA_Request[4]	0 or 1	MAC_TRIG[1]	GMAC1 Auxiliary Timestamp Trigger (MAC1)
ETH_DMA_Request[5]	0 or 1	S3_CONCLK	SERCOS3 Communication synchronized control clock output
ETH_DMA_Request[6]	0 or 1	S3_DIVCLK	SERCOS3 Divided communication clock out

Note 1. Refer to "*DMA Request Allocation*" in the *RZ/N1D Group, RZ/N1S Group, RZ/N1L Group User's Manual: System Control and Peripheral*.

CAUTION

Please set DMAC in "DMAC Flow Controller Mode".

Section 11 MDIO Interfaces

11.1 Overview

The application sends the control data to the PHY and receives status information from the PHY through the MDIO interface as shown in figure below.

The Management Data input/output (MDIO) interface allows the application to access any PHY registers through 2 wires. 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. Only one register in one PHY can be addressed at any given time. For more details about the communication from the Application to the PHYs, refer to *the Reconciliation Sublayer and Media Independent Interface Specifications* section of *the IEEE 802.3z specification, 1000BASE Ethernet*.

Signals are multiplexed with other peripheral IO on GPIO pins.

- Two independent MDIO interfaces MDIO1 & 2
- Each MDIO interface can be driven by following module:
  - GMAC1 & GMAC2
  - HW-RTOS GMAC
  - EtherCAT
  - Sercos III
  - A5PSW
- 1 output mode is managed:
  - Two wires, allows direct connection to PHYs controlled by:
    MDC[2:1], MDIO[2:1]

Figure 11.1 MDIO Interface and Connections

11.2 Signal Interface

Table 11.1 MDIO Signal Interface

Signal Name	I/O	Description	Active
MDC[1]	O	Management data clock	—
MDC[2]	O	Management data clock	—
MDIO[1]	I/O	Management data I/O	—
MDIO[2]	I/O	Management data I/O	—

11.3 Operation

Two independent MDIO interfaces MDIO1 & 2 are managed inside component:

- MDIO1 interface is configuration by bGPIOs_Level2_Config_MDIO1 bit.
- MDIO2 interface is configuration by bGPIOs_Level2_Config_MDIO2 bit.

Figure 11.2 General View of MDIO Multiplexing

The MDIO1 & 2 interface are routed only on 3.3V IO.

- The mode is directly managed by Function 4 of IO Multiplexing Level2 table, see rGPIOs_Level2_Config_[n] with n = 0..169.

Figure 11.3 General View of MDIO Multiplexing

REVISION HISTORY	RZ/N1D Group, RZ/N1S Group, RZ/N1L Group User's Manual: R-IN Engine and Ethernet Peripherals

Rev.	Date	Description	
		Page	Summary
0.50	Jun 30, 2017	—	First Edition issued
0.80	Oct 31, 2017	—	Section 3 and 9: Moved clause "HW-RTOS ACS (Accessory) Register Map" into section "RIN Engine Accessory Register".
		—	Section 4: Fixed bit assignement of "Fig 4.18 VLAN Resolution Table Overview".
		—	Section 4: Merged "system clock" and "swicth clock" into "switch system clock".
		—	Section 4: Merged "cpu_clk" and "reg_clk" into "AHB Clock".
0.90	Dec 28, 2017	—	Section 3: Changed interrupt symbol name.
		—	Section 4: Added a restcition of TDMA function.
		—	Section 5: Change initialization flow to avoid inifinite loop in case that EEPROM is blank.
		—	Section 5: Added a Reset Circuit description to explain reset operation when a reset request from EtherCAT master is detected.
0.95	Oct 19, 2018	—	Fixed typo and expressions in whole document.
		—	Added or modified signals related to PTP in each section. Section 4: Changed ts_ns_in to A5PSW_TS_NS_IN Section 6: Changed clk_ptp_ref_i to GMAC_PTP_REFCLK_I, added GMAC1_PTP_TIMESTAMP_O, GMAC2_PTP_TIMESTAMP_I, Section 7: Added HSR_OVERWRITE_TIME_I
		—	Section 3: Added buffer RAM error description.
		—	Section 4: Removed the word 'Powerlink' from interrupt name.
		—	Section 4: Removed unused register or added note to it. Register: DLR_TX_STAT0/1
		—	Section 4: Added restriction of registers. Register: TDMA_T2/T3, PRIORITY_TYPE1/2, AUTH_POT, COMMAND_CONFIG, IMC_STATUS
		—	Section 4: Added description of registers. Register: MGMT_CONFIG, MODE_CONFIG, COMMAND_CONFIG
		—	Section 6: Removed RevMII registers because it is not supported.
		—	Section 6: Removed RWKPFE bit of PMT_Control_Status register.
		—	Section 6: Added description of enhanced descriptor.
		—	Section 6: Added descriptions of filter command, CRC-16 calculation, and condition of interrupt generation, in Power Management Block.
		—	Section 6: Added description of enhanced descriptor.
		—	Section 8: Fixed type in Figure 8.11.
1.00	Mar 29, 2019	—	Section 4, Section 10: Changed the description of PHYSPEED bit of STATUS_P[n] register of A5PSW. Relating this, also changed the descripton of ENA_10 bit of COMMAND_CONFIG_P[n] register and SET10 bit of SWCTRL register.
		—	Section 4: Changed the number of entry table from 2048 to 8192 in the example for PRP_AGETIME register setting.
		—	Section 4: Added the PHY requirement when half-duplex mode in the restriction.
		—	Section 7: Added how to initialize Duplicate Detection RAM in HSR initialization flow. Also, added the supplement that the initialization flow requires the reset to HSR_CLK50 domain.
1.10	Jul 29, 2019	356	Section 4: Description modified, restriction about TDMA scheduler
		537, 538	Section 6: Description modified, Bus_Mode — Bus Mode Register (PBLx8, RPBL, PBL)
1.20	Feb 28, 2021	—	Fixed typo and expressions in whole document.
		31	Section 1: 1.1 Overview, Table 1.2 Ethernet Peripherals for RZ/N1, description modified

Rev.	Date	Description	
		Page	Summary
1.20	Feb 28, 2021	77	Section 3: 3.5.2.3 MAC DMA Controller, (3) DMA for the Transmission MAC, (d) List of Hardware Function Calls, description modified
		79	Section 3: 3.5.2.4 Buffer RAM DMA Controller, (2) DMA Transfer, (d) List of Hardware Function Calls, description modified
		116	Section 4: 4.1 Overview, description modified
		146	Section 4: 4.4.34 IMC_ERR_FULL — Input Port Memory Full and Truncation Indicator, ipc_err_trunc, ipc_err_full, description added
		147	Section 4: 4.4.35 IMC_ERR_IFACE — Input Port Memory Error Indicator, wbuf_oflow, ipc_err_iface, description added
		148	Section 4: 4.4.36 IMC_ERR_QOFLOW — Output Port Queue Overflow Indicator, op_error, description added
		180	Section 4: 4.4.70 PORT[n]_CTRL — PORT[n] Timestamp Control/Status (n = 0..3), TS_KEEP, description modified
		194	Section 4: 4.4.85 PTPAutoResponse_P[n] — PORT[n] PTP Auto Response Register (n = 0..3), PortNumber1, PortNumber0, description modified
		241	Section 4: 4.4.160 PRP_CONFIG — PRP Configuration Register, RX_DUP_ACCEPT, description modified
		245	Section 4: 4.4.166 PRP_IRQ_CONTROL — PRP Interrupt Control Register, WRONGLAN, description modified
		246	Section 4: 4.4.167 PRP_IRQ_STAT_ACK — PRP Interrupt Status/ACK Register, WRONGLAN, description modified
		249	Section 4: 4.4.172 CntErrWrongLanA — PRP Wrong ID LAN-A Count Register, description modified
		250	Section 4: 4.4.173 CntErrWrongLanB — PRP Wrong ID LAN-B Count Register, description modified
		297	Section 4: 4.5.3.5 Frame Classification and Priority Resolution, (3) IPv4 and IPv6 Priority Look Up, description modified
		300	Section 4: 4.5.3.7 Layer 2 Look Up Engine, (4) Address Memory, description modified
		326	Section 4: 4.5.7.2 Configuration Settings, description modified
		341, 342	Section 4: 4.5.12.2 A5PSW Function Summary, description delete
		346	Section 4: 4.5.12.4 MAC Address Learning Enhancements, Table 4.243 PRP Support Relevant Registers, PRP_GROUP, description modified
		350	Section 4: 4.5.13.3 Hub Specific Forwarding Rules, description modified
		350	Section 4: 4.5.13.4 HUB Group Clocking, description modified
		357, 358	Section 4: 4.6.1 Restriction, description added
		363	Section 5: 5.2 Signal Interface, Table 5.2 Signal Interface of the EtherCAT Slave Controller (excluding PHY MII pins), ETHCAT_WDT_Int, description modified
		483	Section 6: 6.4.17 MAC_Address[n]_High — MAC Address [n] High Register (n = 1..17), Address:, description modified
		484	Section 6: 6.4.18 MAC_Address[n]_Low — MAC Address [n] Low Register (n = 1..17), Address:, description modified
		583	Section 6: 6.5.11.2 Receive Descriptor, Table 6.129 Receive Descriptor Fields 2 (RDES2), table added
		583	Section 6: 6.5.11.2 Receive Descriptor, Table 6.130 Receive Descriptor Fields 3 (RDES3), description modified
1.30	Dec 29, 2021	—	Fixed typo and expressions in whole document.
		105	Section 3: 3.6.3 Management TAG Control, subsection added
		163	Section 4: 4.4.50 AUTH_PORT[n] — PORT[n] Authentication Control and Configuration (n = 0..4), EAPOL_enable, description added
		189	Section 4: 4.4.78 COMMAND_CONFIG_P[n] — Port[n] Command Configuration Register (n = 0..4), ENA_10, description modified

Rev.	Date	Description	
		Page	Summary
1.30	Dec 29, 2021	195	Section 4: 4.4.86 STATUS_P[n] — PORT[n] Port Status Register (n = 0..4), PHYSPEED, description modified
		294	Section 4: 4.5.3.4 Frame Snooping, Figure 4.10 Snooping Arithmetic Function, figure modified
		301	Section 4: 4.5.3.8 Layer 2 Look Up Engine Operational Description, (2) Memory Write, description modified
		324	Section 4: 4.5.6.1 Overview, Figure 4.21 MAC Extension for Autonomous EEE Operation, figure modified
		330	Section 4: 4.5.9 Interrupts, Table 4.238 Interrupt Sources, expression modified
		355	Section 4: 4.5.16 Initializing A5PSW, description modified
		355	Section 4: 4.5.16 Initializing A5PSW, Figure 4.33 Initializing of A5PSW flowchart, figure modified
		450	Section 5: 5.5.1 Initializing, Figure 5.2 Initializing Flowchart, Note 3 deleted
		771	Section 10: 10.2.14 SWCTRL — A5PSW Control Register, SET10, description modified
		772	Section 10: 10.2.16 RMTAGCTRL — HW-RTOS GMAC Management TAG Control Register, MGMT_ENB, note added
1.40	Oct 31,2025	609	7.4.1 RCI_WCONFIG — Send Config Register, bit R/W attribute changed to W except reserved bit, register description added
		613	7.4.6 R_MACADH — Own MAC Address High Register, MLE description modified
		618	7.4.12 R_MACFLT_I[m]L — Filter MAC Address Interlink [m] Low Register (m = 1 to 8), FML description added 7.4.13 R_MACFLT_I[m]H — Filter MAC Address Interlink [m] High Register (m = 1 to 8), FMH description added
		619	7.4.14 R_MACFLT_C[m]L — Filter MAC Address CPU [m] Low Register (m = 1 to 6), FML description added 7.4.15 R_MACFLT_C[m]H — Filter MAC Address CPU [m] High Register (m = 1 to 6), FMH description added
		621	7.4.19 R_FRA_ALL_ARX — Received Frame A Register, ALL description modified
		623	7.4.22 R_FRA_ERR_ARX — Received Error Frame A Register, ERR description modified
		624	7.4.24 R_FRA_ALL_ATX — Sent Frame A Register, ALL description modified 7.4.25 R_FRA_TAG_ATX — Sent Tagged Frame A Register, TAG description modified
		625	7.4.26 R_FRA_NLL_ATX — Sent Not Linklocal Frame A Register, NLL description added 7.4.27 R_FRA_ALL_BRX — Received Frame B Register, ALL description modified
		627	7.4.30 R_FRA_ERR_BRX — Received Error B Register, ERR description modified
		628	7.4.32 R_FRA_ALL_BTX — Sent Frame B Register, ALL description modified 7.4.33 R_FRA_TAG_BTX — Sent Tagged Frame B Register, TAG description modified
		629	7.4.34 R_FRA_NLL_BTX — Sent Not Linklocal Frame B Register, NLL description added 7.4.35 R_FRA_ALL_CRX — Received Frame C Register, ALL description modified
		631	7.4.38 R_FRA_ERR_CRX — Received Error C Register, ERR description modified
		632	7.4.40 R_FRA_ALL_CTX — Sent Frame C Register, ALL description modified 7.4.41 R_FRA_TAG_CTX — Sent Tagged Frame C Register, TAG description modified
		633	7.4.42 R_FRA_NLL_CTX — Sent Not Linklocal Frame C Reg, NLL description added
		634	7.4.44 R_DBG_RPT1 — Internal Debug Reports Register 1, note added
		635,636	7.4.45 R_DBG_RPT2 — Internal Debug Reports Register 2, note added
		640	7.4.51 RPTP_GPO — General Purpose Register, RPT description added
		641	7.4.52 RPTP_INT — PTP Interrupt Register, register description added
		643	7.4.54 RPTP_TS_STAT_[m] — Timestamp Status Register P[m] (m = 1 to 4), TRP description modified, register description added
		644	7.4.55 RPTP_TS_RD_[m] — Timestamp Read Register P[m] (m = 1 to 4), register description added

Rev.	Date	Description	
		Page	Summary
1.40	Oct 31,2025	645	7.4.56 RPTP_PORT_CONF_[m] — Port Config Register P[m] (m = 1 to 4), PDP description modified, register description added 7.4.57 RPTP_P_DELAY_[m] — Peer Delay Register P[m] (m = 1 to 4). PED description modified & bit R/W attribute changed to W, register description added
		646	7.4.58 RPTP_PHY_DLY_TX_[m] — PHY Tx Delay Register P[m] (m = 1 to 4), register description added 7.4.59 RPTP_PHY_DLY_RX_[m] — PHY Rx Delay Register P[m] (m = 1 to 4), register description added
		647	7.4.60 RPTP_BUF_STAT — Buffer Status Register, register description added
		648	7.5.1.1 Initializing, information for PTP added.

RZ/N1D Group, RZ/N1S Group, RZ/N1L Group
User's Manual: R-IN Engine and Ethernet Peripherals

Publication Date:	Rev.0.50	Jun 30, 2017
	Rev.1.40	Oct 31, 2025

Published by: Renesas Electronics Corporation

RZ/N1D Group, RZ/N1S Group, RZ/N1L Group

Renesas Electronics Corporation

R01UH0753EJ0140