Chapter 41
10/100-Mbps Ethernet MAC (ENET)

41.1 Chip-specific ENET information

Table 41-1. Reference links to related information

Topic	Related module(s)	Reference
System memory map	-	System Memory Map
Clocking	CCM	Clock Management Clock Control Module (CCM)
Power management	PMU	Power Management Power Management Unit
Signal multiplexing	IOMUX	External Signals and Pin Multiplexing IOMUX
Interrupts, DMA Events and XBAR Assignments	-	Interrupts, DMA Events and XBAR Assignments

41.2 Overview

The core implements a dual-speed 10/100-Mbit/s Ethernet MAC compliant with the IEEE802.3-2002 standard. The MAC layer provides compatibility with half- or full-duplex 10/100-Mbit/s Ethernet LANs.

The MAC operation is fully programmable and can be used in Network Interface Card (NIC), bridging, or switching applications. The core implements the remote network monitoring (RMON) counters according to IETF RFC 2819.

The core also implements a hardware acceleration block to optimize the performance of network controllers providing TCP/IP, UDP, and ICMP protocol services. The acceleration block performs critical functions in hardware, which are typically implemented with large software overhead.

The core implements programmable embedded FIFOs that can provide buffering on the receive path for lossless flow control.

Advanced power management features are available with magic packet detection and programmable power-down modes.

A unified DMA (uDMA), internal to the ENET module, optimizes data transfer between the ENET core and the SoC, and supports an enhanced buffer descriptor programming model to support IEEE 1588 functionality.

The programmable Ethernet MAC with IEEE 1588 integrates a standard IEEE 802.3 Ethernet MAC with a time-stamping module. The IEEE 1588 standard provides accurate clock synchronization for distributed control nodes for industrial automation applications.

41.2.1 Block diagram

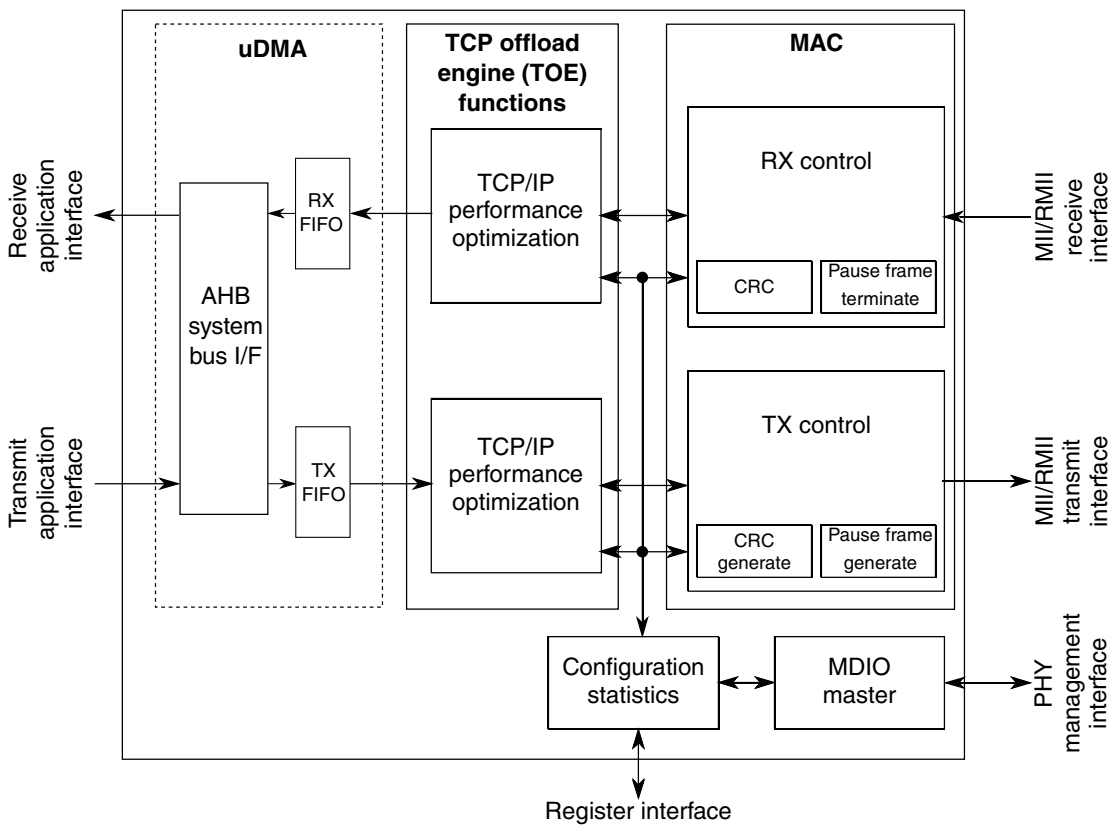


Figure 41-1. Ethernet MAC-NET core block diagram

41.2.2 Features

The MAC-NET core includes the following features.

41.2.2.1 Ethernet MAC features

- Implements the full 802.3 specification with preamble/SFD generation, frame padding generation, CRC generation and checking
- Supports zero-length preamble
- Dynamically configurable to support 10/100-Mbit/s operation
- Supports 10/100 Mbit/s full-duplex and configurable half-duplex operation
- Compliant with the AMD magic packet detection with interrupt for node remote power management
- Seamless interface to commercial ethernet PHY devices via one of the following:
  - a 4-bit Media Independent Interface (MII) operating at 2.5/25 MHz.
  - a 4-bit non-standard MII-Lite (MII without the CRS and COL signals) operating at 2.5/25 MHz.
  - a 2-bit Reduced MII (RMII) operating at 50 MHz.
- CRC-32 checking at full speed with optional forwarding of the frame check sequence (FCS) field to the client
- CRC-32 generation and append on transmit or forwarding of user application provided FCS selectable on a per-frame basis
- In full-duplex mode:
  - Supports Ethernet pause frame (802.3 Annex 31A) providing fully automated flow control without any user application overhead
  - Pause quanta used to form pause frames, dynamically programmable
  - Pause frame generation additionally controllable by user application offering flexible traffic flow control
  - Optional forwarding of received pause frames to the user application
  - Implements standard flow-control mechanism
- In half-duplex mode: provides full collision support, including jamming, backoff, and automatic retransmission
- Supports VLAN-tagged frames according to IEEE 802.1Q
- Programmable MAC address: Insertion on transmit; discards frames with mismatching destination address on receive (except broadcast and pause frames)
- Programmable promiscuous mode support to omit MAC destination address checking on receive
- Multicast and unicast address filtering on receive based on 64-entry hash table, reducing higher layer processing load
- Programmable frame maximum length providing support for any standard or proprietary frame length
- Statistics indicators for frame traffic and errors (alignment, CRC, length) and pause frames providing for IEEE 802.3 basic and mandatory management information database (MIB) package and remote network monitoring (RFC 2819)

- Simple handshake user application FIFO interface with fully programmable depth and threshold levels
- Provides separate status word for each received frame on the user interface providing information such as frame length, frame type, VLAN tag, and error information
- Multiple internal loopback options
- Programmable Clause 22 and Clause 45 MDIO Master interface for PHY device configuration and management
- Supports legacy FEC buffer descriptors
- Supports interupt coalescing

41.2.2.2 IP protocol performance optimization features

- Operates on TCP/IP and UDP/IP and ICMP/IP protocol data or IP header only
- Enables wire-speed processing
- Supports IPv4 and IPv6
- Transparent passing of frames of other types and protocols
- Supports VLAN tagged frames according to IEEE 802.1q with transparent forwarding of VLAN tag and control field
- Automatic IP-header and payload (protocol specific) checksum calculation and verification on receive
- Automatic IP-header and payload (protocol specific) checksum generation and insertion on transmit. Configurable on a per-frame basis
- Supports IP and TCP, UDP, ICMP data for checksum generation and checking
- Supports full header options for IPv4 and TCP protocol headers
- Provides IPv6 support to datagrams with base header only — datagrams with extension headers are passed transparently unmodifed/unchecked
- Provides statistics information for received IP and protocol errors
- Configurable automatic discard of erroneous frames
- Configurable automatic host-to-network (RX) and network-to-host (TX) byte order conversion for IP and TCP/UDP/ICMP headers within the frame
- Configurable padding remove for short IP datagrams on receive

- Configurable Ethernet payload alignment to allow for 32-bit word-aligned header and payload processing
- Programmable store-and-forward operation with clock and rate decoupling FIFOs

41.2.2.3 IEEE 1588 features

- Supports all IEEE 1588 frames.
- Allows reference clock to be chosen independently of network speed.
- Software-programmable precise time-stamping of ingress and egress frames
- Timer monitoring capabilities for system calibration and timing accuracy management
- Precise time-stamping of external events with programmable interrupt generation
- Programmable event and interrupt generation for external system control
- Supports hardware- and software-controllable timer synchronization.
- Provides a 4-channel IEEE 1588 timer. Each channel supports input capture and output compare using the 1588 counter.

41.3 Functional description

This section provides a complete functional description of the MAC-NET core.

41.3.1 Ethernet MAC frame formats

- Minimum length of 64 bytes
- Maximum length of 1518 bytes excluding the preamble and the start frame delimiter (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)

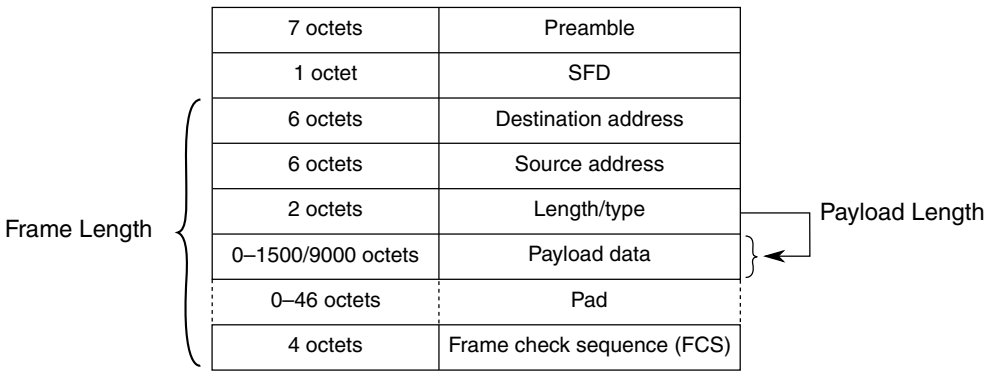


Figure 41-2. MAC frame format overview

Optionally, MAC frames can be VLAN-tagged with an additional four-byte field 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, excluding the preamble and the SFD bytes.

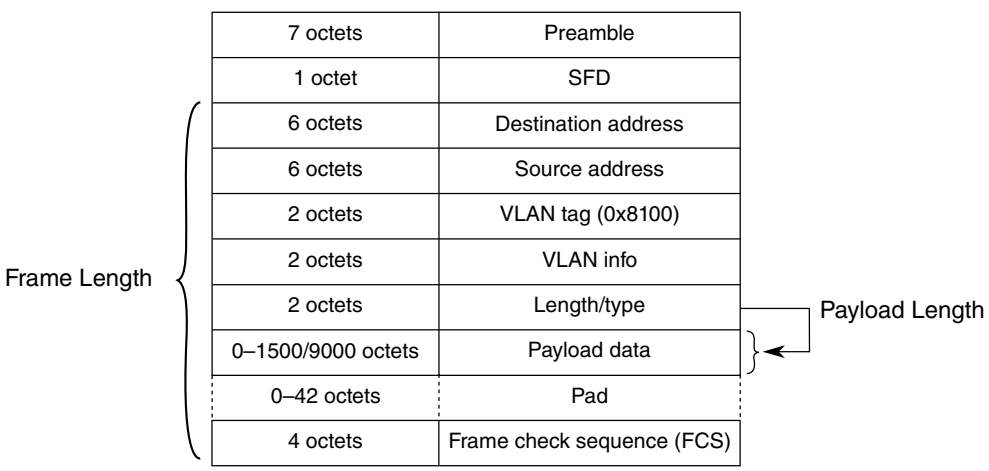


Figure 41-3. VLAN-tagged MAC frame format overview

Table 41-2. MAC frame definition

Term	Description
Frame length	Defines the length, in octets, 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.
Payload length	The length/type field indicates the length of the frame's payload section. The most significant byte is sent/received first.

Table continues on the next page...

Table 41-2. MAC frame definition (continued)

Term	Description
	• If the length/type field is set to a value less than 46, 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 than the programmed frame maximum length (e.g. 1518) 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 first two least significant bits of the MAC address distinguish MAC frames, as detailed in MAC address check.

Note

Although the IEEE specification defines a maximum frame length, the MAC core provides the flexibility to program any value for the frame maximum length.

41.3.1.1 Pause Frames

The receiving device generates a pause frame to indicate a congestion to the emitting device, which should stop sending data.

Pause frames are indicated by the length/type set to 0x8808. 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 (P1) and 3 (P2) as defined in the following table. The P1 pause quanta byte is the most significant.

Table 41-3. 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										
26	6B	AE	0A										
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 than 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.

If a pause frame with a pause value of zero (XON condition) is received, the transmitter is allowed to send data immediately (see Full-duplex flow control operation for details).

41.3.1.2 Magic packets

A magic packet is a unicast, multicast, or broadcast packet, which carries a defined sequence in the payload section.

Magic packets are received and inspected only under specific conditions as described in Magic packet detection.

The defined sequence to decode a magic packet is formed with a synchronization stream which consists of six consecutive 0xFF bytes, and is followed by sequence of sixteen consecutive unicast MAC addresses of the node to be awakened.

This sequence can be located anywhere in the magic packet payload. The magic packet is formed with a standard Ethernet header, optional padding, and CRC.

41.3.2 IP and higher layers frame format

The following sections use the term datagram to describe the protocol specific data unit that 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.

41.3.2.1 Ethernet types

IP datagrams are carried in the payload section of an Ethernet frame. The Ethernet frame type/length field discriminates several datagram types.

The following table lists the types of interest:

Table 41-4. Ethernet type value examples

Type	Description
0x8100	VLAN-tagged frame. The actual type is found 4 octets later in the frame.

Table continues on the next page...

Table 41-4. Ethernet type value examples (continued)

Type	Description
0x0800	IPv4
0x0806	ARP
0x86DD	IPv6

41.3.2.2 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 first byte sent/received is the leftmost byte of the first word (in other words, 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 immediately followed by the payload, which can contain further protocol headers (for example, TCP or UDP, as indicated by the protocol field value). The complete IP datagram is transported in the payload section of an Ethernet frame.

Table 41-5. IPv4 header 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
Version	IHL	TOS		Length			
Fragment ID				Flags	Fragment offset		
TTL		Protocol		Header checksum			
Source address							
Destination address							
Options							

Table 41-6. IPv4 header fields

Field name	Description
Version	4-bit IP version information. 0x4 for IPv4 frames.
IHL	4-bit Internet header length information. Determines number of 32-bit words found within the IP header. If no options are present, the default value is 0x5.
TOS	Type of service/DiffServ field.
Length	Total length of the datagram in bytes, including all octets of header and payload.
Fragment ID, flags, fragment offset	Fields used for IP fragmentation.
TTL	Time-to-live. In effect, is decremented at each router arrival. If zero, datagram must be discarded.
Protocol	Identifier of protocol that follows in the datagram.

Table continues on the next page...

Table 41-6. IPv4 header fields (continued)

Field name	Description
Header checksum	Checksum of IP header. For computational purposes, this field's value is zero.
Source address	Source IP address.
Destination address	Destination IP address.

41.3.2.3 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 ten words (40 bytes). The next header field identifies the type of the header that follows the IPv6 header. It is defined similar to the protocol identifier within IPv4, with new definitions for identifying extension headers. These headers can be inserted between the IPv6 header and the protocol header, which will shift the protocol header accordingly.The accelerator currently only supports IPv6 without extension headers (in other words, the next header specifies TCP, UDP, or IMCP).

The first byte sent/received is the leftmost byte of the first word (in other words, version/ traffic class fields).

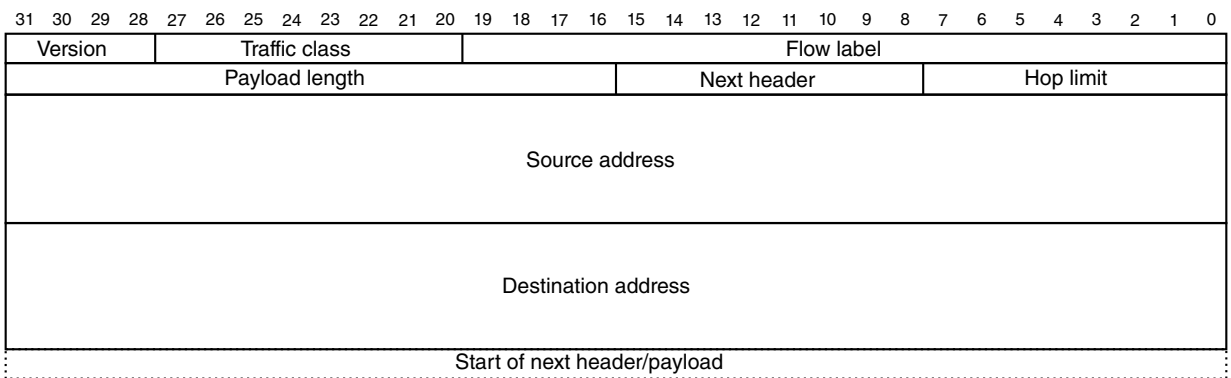


Figure 41-4. IPv6 header format

Table 41-7. IPv6 header fields

Field name	Description
Version	4-bit IP version information. 0x6 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.

Table continues on the next page...

Table 41-7. IPv6 header fields (continued)

Field name	Description
Hop limit	Hop counter, decremented by one 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.

41.3.2.4 Internet Control Message Protocol (ICMP) datagram format

An internet control message protocol (ICMP) is found following the IP header, if the protocol identifier is 1. The ICMP datagram has a four-octet header followed by additional message data.

Table 41-8. ICMP header 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
Type		Code		Checksum			
ICMP message data							

Table 41-9. IP header fields

Field name	Description
Type	8-bit type information
Code	8-bit code that is related to the message type
Checksum	16-bit one's complement checksum over the complete ICMP datagram

41.3.2.5 User Datagram Protocol (UDP) datagram format

A user datagram protocol header is found after the IP header, when the protocol identifier is 17.

The payload of the datagram is after the UDP header. The header byte order follows the conventions given for the IP header above.

Table 41-10. UDP header 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
Source port				Destination port			
Length				Checksum			

Table 41-11. UDP header fields

Field name	Description
Source port	Source application port
Destination port	Destination application port
Length	Length of user data which immediately follows the header, including the UDP header (that is, minimum value is 8)
Checksum	Checksum over the complete datagram and some IP header information

41.3.2.6 TCP datagram format

A TCP header is found following the IP header, when the protocol identifier has a value of 6.

The TCP payload immediately follows the TCP header.

Table 41-12. TCP header 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
Source port				Destination port			
Sequence number							
Acknowledgement number							
Offset	Reserved		Flags	Window			
Checksum				Urgent pointer			
Options							

Table 41-13. TCP header fields

Field name	Description
Source port	Source application port
Destination port	Destination application port
Sequence number	Transmit sequence number
Ack. number	Receive sequence number
Offset	Data offset, which is number of 32-bit words within TCP header — if no options selected, defaults to value of 5
Flags	URG, ACK, PSH, RST, SYN, FIN flags
Window	TCP receive window size information
Checksum	Checksum over the complete datagram (TCP header and data) and IP header information
Options	Additional 32-bit words for protocol options

41.3.3 IEEE 1588 message formats

The following sections describe the IEEE 1588 message formats.

41.3.3.1 Transport encapsulation

The precision time protocol (PTP) datagrams are encapsulated in Ethernet frames using the UDP/IP transport mechanism, or optionally, with the newer 1588v2 directly in Ethernet frames (layer 2).

Typically, multicast addresses are used to allow efficient distribution of the synchronization messages.

41.3.3.1.1 UDP/IP

The 1588 messages (v1 and v2) can be transported using UDP/IP multicast messages.

The table below shows IP multicast groups defined for PTP. The table also shows their respective MAC layer multicast address mapping according to RFC 1112 (last three octets of IP follow the fixed value of 01-00-5E).

Table 41-14. UDP/IP multicast domains

Name	IP Address	MAC Address mapping
DefaultPTPdomain	224.0.1.129	01-00-5E-00-01-81
AlternatePTPdomain1	224.0.1.130	01-00-5E-00-01-82
AlternatePTPdomain2	224.0.1.131	01-00-5E-00-01-83
AlternatePTPdomain3	224.0.1.132	01-00-5E-00-01-84

Table 41-15. UDP port numbers

Message type	UDP port	Note
Event	319	Used for SYNC and DELAY_REQUEST messages
General	320	All other messages (for example, follow-up, delay-response)

41.3.3.1.2 Native Ethernet (PTPv2)

In addition to using UDP/IP frames, IEEE 1588v2 defines a native Ethernet frame format that uses ethertype = 0x88F7. The payload of the Ethernet frame immediately contains the PTP datagram, starting with the PTPv2 header.

Besides others, version 2 adds a peer delay mechanism to allow delay measurements between individual point-to-point links along a path over multiple nodes. The following multicast domains are also defined in PTPv2.

Table 41-16. PTPv2 multicast domains

Name	MAC address
Normal messages	01-1B-19-00-00-00
Peer delay messages	01-80-C2-00-00-0E

41.3.3.2 PTP header

All PTP frames contain a common header that determines the protocol version and the type of message, which defines the remaining content of the message.

All multi-octet fields are transmitted in big-endian order (the most significant byte is transmitted/received first).

The last four bits of versionPTP are at the same position (second byte) for PTPv1 and PTPv2 headers. This allows accurate identification by inspecting the first two bytes of the message.

41.3.3.2.1 PTPv1 header

Table 41-17. Common PTPv1 message header

Offset	Octets	Bits							
		7	6	5	4	3	2	1	0
0	2	versionPTP = 0x0001							
2	2	versionNetwork							
4	16	subdomain							
20	1	messageType							
21	1	sourceCommunicationTechnology							
22	6	sourceUuid							
28	2	sourcePortId							
30	2	sequenceId							
32	1	control							
33	1	0x00							
34	2	flags							
36	4	reserved							

The type of message is encoded in the messageType and control fields as shown in Table 41-18 :

Table 41-18. PTPv1 message type identification

messageType	control	Message Name	Message
0x01	0x0	SYNC	Event message
0x01	0x1	DELAY_REQ	Event message
0x02	0x2	FOLLOW_UP	General message
0x02	0x3	DELAY_RESP	General message
0x02	0x4	MANAGEMENT	General message
other	other	—	Reserved

The field sequenceId is used to non-ambiguously identify a message.

41.3.3.2.2 PTPv2 header

Table 41-19. Common PTPv2 message header

Offset	Octets	Bits							
		7	6	5	4	3	2	1	0
0	1	transportSpecific				messageId			
1	1	reserved				versionPTP = 0x2			
2	2	messageLength							
4	1	domainNumber							
5	1	reserved							
6	2	flags							
8	8	correctionField							
16	4	reserved							
20	10	sourcePortIdentity							
30	2	sequenceId							
32	1	control							
33	1	logMeanMessageInterval							

The type of message is encoded in the field messageId as follows:

Table 41-20. PTPv2 message type identification

messageId	Message name	Message
0x0	SYNC	Event message
0x1	DELAY_REQ	Event message
0x2	PATH_DELAY_REQ	Event message
0x3	PATH_DELAY_RESP	Event message
0x4–0x7	—	Reserved
0x8	FOLLOW_UP	General message
0x9	DELAY_RESP	General message

Table continues on the next page...

Table 41-20. PTPv2 message type identification (continued)

messageId	Message name	Message
0xa	PATH_DELAY_FOLLOW_UP	General message
0xb	ANNOUNCE	General message
0xc	SIGNALING	General message
0xd	MANAGEMENT	General message

The PTPv2 flags field contains further details on the type of message, especially if one-step or two-step implementations are used. The one- or two-step implementation is controlled by the TWO_STEP bit in the first octet of the flags field as shown below. Reserved bits are cleared.

Table 41-21. PTPv2 message flags field definitions

Bit	Name	Description
0	ALTERNATE_MASTER	See IEEE 1588 Clause 17.4
1	TWO_STEP	1 Two-step clock 0 One-step clock
2	UNICAST	1 Transport layer address uses a unicast destination address 0 Multicast is used
3	—	Reserved
4	—	Reserved
5	Profile specific	
6	Profile specific	
7	—	Reserved

41.3.4 MAC receive

The MAC receive engine performs the following tasks:

- Check frame framing
- Remove frame preamble and frame SFD field
- Discard frame based on frame destination address field
- Terminate pause frames
- Check frame length
- Remove payload padding if it exists

- Calculate and verify CRC-32
- Write received frames in the core receive FIFO

If the MAC is programmed to operate in half-duplex mode, it will also check if the frame is received with a collision.

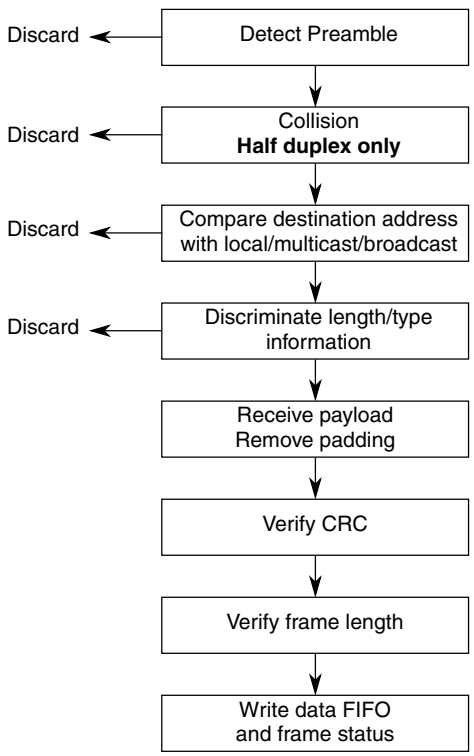


Figure 41-5. MAC receive flow

41.3.4.1 Collision detection in half-duplex mode

If the packet is received with a collision detected during reception of the first 64 bytes, the packet is discarded (if frame size was less than ~14 octets) or transmitted to the user application with an error and RxBD[CE] set.

41.3.4.2 Preamble processing

The IEEE 802.3 standard allows a maximum size of 56 bits (seven bytes) for the preamble, while the MAC core allows any preamble length, including zero length preamble.

The MAC core checks for the start frame delimiter (SFD) byte. If the next byte of the preamble, which is different from 0x55, is not 0xD5, the frame is discarded.

Although the IEEE specification dictates that the inner-packet gap should be at least 96 bits, the MAC core is designed to accept frames separated by only 64 10/100-Mbit/s operation (MII) bits.

The MAC core removes the preamble and SFD bytes.

41.3.4.3 MAC address check

The destination address bit 0 differentiates between multicast and unicast addresses.

- If bit 0 is 0, the MAC address is an individual (unicast) address.
- If bit 0 is 1, the MAC address defines a group (multicast) address.
- If all 48 bits of the MAC address are set, it indicates a broadcast address.

41.3.4.3.1 Unicast address check

If a unicast address is received, the destination MAC address is compared to the node MAC address programmed by the host in the PADDR1/2 registers.

If the destination address matches any of the programmed MAC addresses, the frame is accepted.

If Promiscuous mode is enabled (RCR[PROM] = 1) no address checking is performed and all unicast frames are accepted.

41.3.4.3.2 Multicast and unicast address resolution

The hash table algorithm used in the group and individual hash filtering operates as follows.

- The 48-bit destination address is mapped into one of 64 bits, represented by 64 bits in ENET*n*_GAUR/GALR (group address hash match) or ENET*n*_IAUR/IALR (individual address hash match).
- This mapping is performed by passing the 48-bit address through the on-chip 32-bit CRC generator and selecting the six most significant bits of the CRC-encoded result to generate a number between 0 and 63.
- The msb of the CRC result selects ENET*n*_GAUR (msb = 1) or ENET*n*_GALR (msb = 0).
- The five lsbs of the hash result select the bit within the selected register.
- If the CRC generator selects a bit set in the hash table, the frame is accepted; else, it is rejected.

For example, if eight group addresses are stored in the hash table and random group addresses are received, the hash table prevents roughly 56/64 (or 87.5%) of the group address frames from reaching memory. Those that do reach memory must be further filtered by the processor to determine if they truly contain one of the eight desired addresses.

The effectiveness of the hash table declines as the number of addresses increases.

The user must initialize the hash table registers. Use this CRC32 polynomial to compute the hash:

- $FCS(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^{8} + x^{7} + x^{5} + x^{4} + x^{2} + x^{1} + 1$

If Promiscuous mode is enabled (ENET*n*_RCR[PROM] = 1) all unicast and multicast frames are accepted regardless of ENET*n*_GAUR/GALR and ENET*n*_IAUR/IALR settings.

41.3.4.3.3 Broadcast address reject

All broadcast frames are accepted if BC_REJ is cleared or ENET*n*_RCR[PROM] is set. If PROM is cleared when ENET*n*_RCR[BC_REJ] is set, all broadcast frames are rejected.

Table 41-22. Broadcast address reject programming

PROM	BC_REJ	Broadcast frames
0	0	Accepted
0	1	Rejected
1	0	Accepted
1	1	Accepted

41.3.4.3.4 Miss-bit implementation

For higher layer filtering purposes, RxBD[M] indicates an address miss when the MAC operates in promiscuous mode and accepts a frame that would otherwise be rejected.

If a group/individual hash or exact match does not occur and Promiscuous mode is enabled (RCR[PROM] = 1), the frame is accepted and the M bit is set in the buffer descriptor; otherwise, the frame is rejected.

This means the status bit is set in any of the following conditions during Promiscuous mode:

- A broadcast frame is received when BC_REJ is set

- A unicast is received that does not match either:
  - Node address (PALR[PADDR1] and PAUR[PADDR2])
  - Hash table for unicast (IAUR[IADDR1] and IALR[IADDR2])
- A multicast is received that does not match the GAUR[GADDR1] and GALR[GADDR2] hash table entries

41.3.4.4 Frame length/type verification: payload length check

If the length/type is less than 0x600 and NLC is set, the MAC checks the payload length and reports any error in the frame status word RCR[NLC] and interrupt bit EIR[PLR].

If the length/type is greater than or equal to 0x600, the MAC interprets the field as a type and no payload length check is performed.

The length check is performed on VLAN and stacked VLAN frames. If a padded frame is received, no length check can be performed due to the extended frame payload because padded frames can never have a payload length error.

41.3.4.5 Frame length/type verification: frame length check

When the receive frame length exceeds MAX_FL bytes, the BABR interrupt is generated and the RxBD[LG] bit is set.

The frame is not truncated unless the frame length exceeds the value programmed in ENET*n*_FTRL[TRUNC_FL]. If the frame is truncated, RxBD[TR] is set. In addition, a truncated frame always has the CRC error indication set (RxBD[CR]).

41.3.4.6 VLAN frames processing

VLAN frames have a length/type field set to 0x8100 immediately followed by a 16-Bit VLAN control information field.

VLAN-tagged frames are received as normal frames because the VLAN tag is not interpreted by the MAC function, and are pushed complete with the VLAN tag to the user application. If the length/type field of the VLAN-tagged frame, which is found four octets later in the frame, is less than 42, the padding is removed. In addition, the frame status word (RxBD[VLAN]) indicates that the current frame is VLAN tagged.

41.3.4.7 Pause frame termination

The receive engine terminates pause frames and does not transfer them to the receive FIFO. The quanta is extracted and sent to the MAC transmit path via a small internal clock rate decoupling asynchronous FIFO.

The quanta is written only if a correct CRC and frame length are detected by the control state machine. If not, the quanta is discarded and the MAC transmit path is not paused.

Valid pause frames are ignored if ENET*n*_RCR[FCE] is cleared and are forwarded to the client interface when ENET*n*_RCR[PAUFWD] is set.

41.3.4.8 CRC check

The CRC-32 field is checked and forwarded to the core FIFO interface if ENET*n*_RCR[CRCFWD] is cleared and ENET*n*_RCR[PADEN] is cleared. When CRCFWD is set (regardless of PADEN), the CRC-32 field is checked and terminated (not transmitted to the FIFO).

The CRC polynomial, as specified in the 802.3 standard, is:

- $FCS(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^{8} + x^{7} + x^{5} + x^{4} + x^{2} + x^{1} + 1$

The 32 bits of the CRC value are placed in the frame check sequence (FCS) field with the x^{31} term as right-most bit of the first octet. The CRC bits are thus received in the following order: x^{31}, x^{30},..., x^{1}, x^{0}.

If a CRC error is detected, the frame is marked invalid and RxBD[CR] is set.

41.3.4.9 Frame padding removal

When a frame is received with a payload length field set to less than 46 (42 for VLAN-tagged frames and 38 for frames with stacked VLANs), the zero padding can be removed before the frame is written into the data FIFO depending on the setting of ENET*n*_RCR[PADEN].

Note

If a frame is received with excess padding (in other words, the length field is set as mentioned above, but the frame has more than 64 octets) and padding removal is enabled, then the

padding is removed as normal and no error is reported if the frame is otherwise correct (for example: good CRC, less than maximum length, and no other error).

41.3.5 MAC transmit

Frame transmission starts when the transmit FIFO holds enough data.

After a transfer starts, the MAC transmit function performs the following tasks:

- Generates preamble and SFD field before frame transmission
- Generates XOFF pause frames if the receive FIFO reports a congestion or if ENET*n*_TCR[TFC_PAUSE] is set with ENET*n*_OPD[PAUSE_DUR] set to a non-zero value
- Generates XON pause frames if the receive FIFO congestion condition is cleared or if TFC_PAUSE is set with PAUSE_DUR cleared
- Suspends Ethernet frame transfer (XOFF) if a non-zero pause quanta is received from the MAC receive path
- Adds padding to the frame if required
- Calculates and appends CRC-32 to the transmitted frame
- Sends the frame with correct inter-packet gap (IPG) (deferring)

When the MAC is configured to operate in half-duplex mode, the following additional tasks are performed:

- Collision detection
- Frame retransmit after back-off timer expires

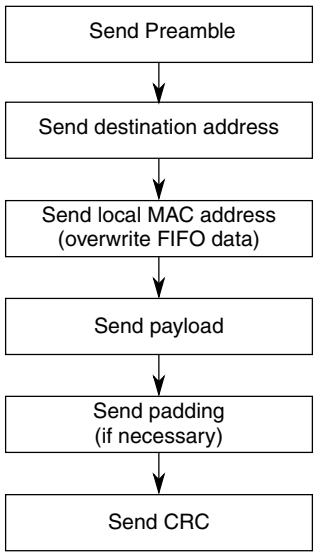


Figure 41-6. Frame transmit overview

41.3.5.1 Frame payload padding

The IEEE specification defines a minimum frame length of 64 bytes.

If the frame sent to the MAC from the user application has a size smaller than 60 bytes, the MAC automatically adds padding bytes (0x00) to comply with the Ethernet minimum frame length specification. Transmit padding is always performed and cannot be disabled.

If the MAC is not allowed to append a CRC (TxBD[TC] = 1), the user application is responsible for providing frames with a minimum length of 64 octets.

41.3.5.2 MAC address insertion

On each frame received from the core transmit FIFO interface, the source MAC address is either:

- Replaced by the address programmed in the PADDR1/2 fields (ENET*n*_TCR[ADDINS] = 1)
- Transparently forwarded to the Ethernet line (ENET*n*_TCR[ADDINS] = 0)

41.3.5.3 CRC-32 generation

The CRC-32 field is optionally generated and appended at the end of a frame.

The CRC polynomial, as specified in the 802.3 standard, is:

- $FCS(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^{8} + x^{7} + x^{5} + x^{4} + x^{2} + x^{1} + 1$

The 32 bits of the CRC value are placed in the FCS field so that the x^{31} term is the right-most bit of the first octet. The CRC bits are thus transmitted in the following order: x^{31}, x^{30},..., x^{1}, x^{0}.

41.3.5.4 Inter-packet gap (IPG)

In full-duplex mode, after frame transmission and before transmission of a new frame, an IPG (programmed in ENET*n*_TIPG) is maintained. The minimum IPG can be programmed between 8 and 26 byte-times (64 and 208 bit-times).

In half-duplex mode, the core constantly monitors the line. Actual transmission of the data onto the network occurs only if it has been idle for a 96-bit time period, and any back-off time requirements have been satisfied. In accordance with the standard, the core begins to measure the IPG from CRS de-assertion.

41.3.5.5 Collision detection and handling — half-duplex operation only

A collision occurs on a half-duplex network when concurrent transmissions from two or more nodes take place. During transmission, the core monitors the line condition and detects a collision when the PHY device asserts COL.

When the core detects a collision while transmitting, it stops transmission of the data and transmits a 32-bit jam pattern. If the collision is detected during the preamble or the SFD transmission, the jam pattern is transmitted after completing the SFD, which results in a minimum 96-bit fragment. The jam pattern is a fixed pattern that is not compared to the actual frame CRC, and has a very low probability (0.532) of having a jam pattern identical to the CRC.

If a collision occurs before transmission of 64 bytes (including preamble and SFD), the MAC core waits for the backoff period and retransmits the packet data (stored in a 64-byte re-transmit buffer) that has already been sent on the line. The backoff period is generated from a pseudo-random process (truncated binary exponential backoff).

If a collision occurs after transmission of 64 bytes (including preamble and SFD), the MAC discards the remainder of the frame, optionally sets the LC interrupt bit, and sets TxBD[LCE].

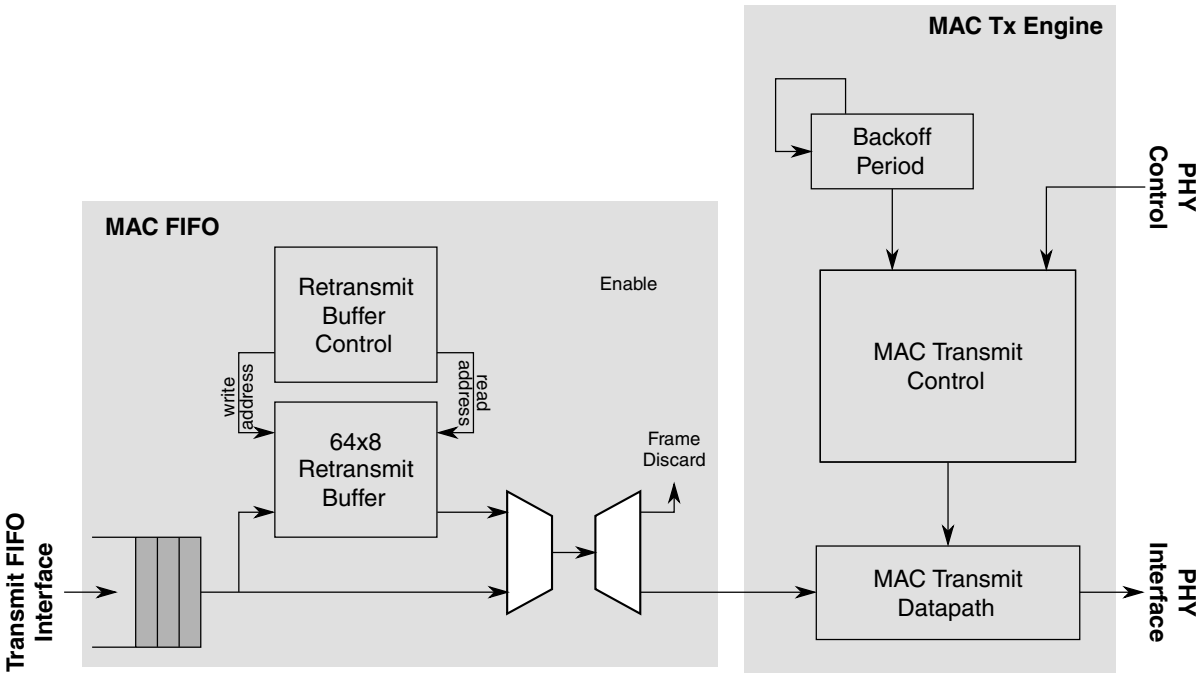


Figure 41-7. Packet re-transmit overview

The backoff time is represented by an integer multiple of slot times. One slot is equal to a 512-bit time period. The number of the delay slot times, before the n^{th} re-transmission attempt, is chosen as a uniformly-distributed random integer in the range:

- $0 < r < 2^k$
- $k = \min(n, N)$; where n is the number of retransmissions and $N = 10$

For example, after the first collision, the backoff period is 0 or 1 slot time. If a collision occurs on the first retransmission, the backoff period is 0, 1, 2, or 3, and so on.

The maximum backoff time (in 512-bit time slots) is limited by N = 10 as specified in the IEEE 802.3 standard.

If a collision occurs after 16 consecutive retransmissions, the core reports an excessive collision condition (ENET*n*_EIR[RL] interrupt field and TxBD[EE]) and discards the current packet from the FIFO.

In networks violating the standard requirements, a collision may occur after transmission of the first 64 bytes. In this case, the core stops the current packet transmission and discards the rest of the packet from the transmit FIFO. The core resumes transmission with the next packet available in the core transmit FIFO.

warning

Ethernet PHYs that support the SQE Test, or "heartbeat,"
feature must disable this feature. When this feature is enabled,

the PHY asserts the collision signal after a frame is transmitted to indicate to the ENET that the PHY's collision logic is working. This may cause data corruption in the next frame from the ENET. This corrupted frame contains up to 21 zero bytes which start somewhere within the MAC destination address field. The ENET, however, will still generate a good FCS (CRC-32) but with corrupted data.

41.3.6 Full-duplex flow control operation

Three conditions are handled by the core's flow control engine:

- Remote device congestion — The remote device connected to the same Ethernet segment as the core reports congestion and requests that the core stop sending data.
- Core FIFO congestion — When the core's receive FIFO reaches a user-programmable threshold (RX section empty), the core sends a pause frame back to the remote device requesting the data transfer to stop.
- Local device congestion — Any device connected to the core can request (typically, via the host processor) the remote device to stop transmitting data.

41.3.6.1 Remote device congestion

When the MAC transmit control gets a valid pause quanta from the receive path and if ENET*n*_RCR[FCE] is set, the MAC transmit logic:

- Completes the transfer of the current frame.
- Stops sending data for the amount of time specified by the pause quanta in 512 bit time increments.
- Sets ENET*n*_TCR[RFC_PAUSE].

Frame transfer resumes when the time specified by the quanta expires and if no new quanta value is received, or if a new pause frame with a quanta value set to 0x0000 is received. The MAC also resets RFC_PAUSE to zero.

If ENET*n*_RCR[FCE] cleared, the MAC ignores received pause frames.

Optionally and independent of ENET*n*_RCR[FCE], pause frames are forwarded to the client interface if PAUFWD is set.

41.3.6.2 Local device/FIFO congestion

The MAC transmit engine generates pause frames when the local receive FIFO is not able to receive more than a pre-defined number of words (FIFO programmable threshold) or when pause frame generation is requested by the local host processor.

- To generate a pause frame, the host processor sets ENET*n*_TCR[TFC_PAUSE]. A single pause frame is generated when the current frame transfer is completed and TFC_PAUSE is automatically cleared. Optionally, an interrupt is generated.
- An XOFF pause frame is generated when the receive FIFO asserts its section empty flag (internal). An XOFF pause frame is generated automatically, when the current frame transfer completes.
- An XON pause frame is generated when the receive FIFO deasserts its section empty flag (internal). An XON pause frame is generated automatically, when the current frame transfer completes.

When an XOFF pause frame is generated, the pause quanta (payload byte P1 and P2) is filled with the value programmed in ENET*n*_OPD[PAUSE_DUR].

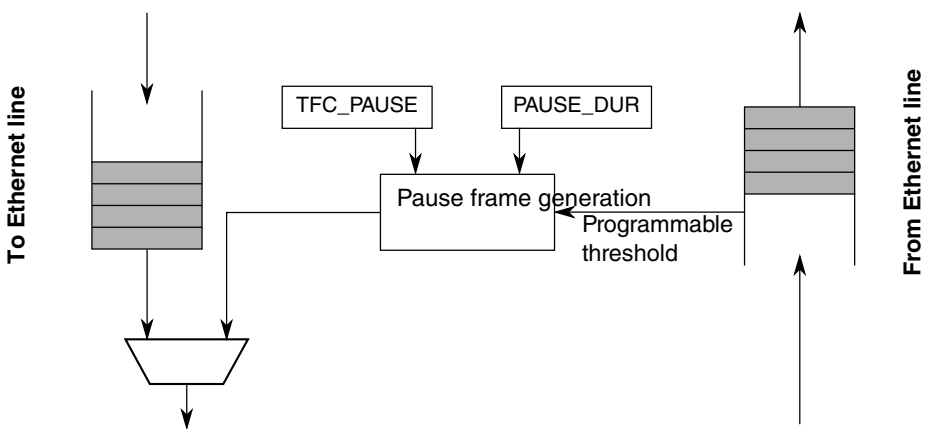


Figure 41-8. Pause frame generation overview

Note

Although the flow control mechanism should prevent any FIFO overflow on the MAC core receive path, the core receive FIFO is protected. When an overflow is detected on the receive FIFO, the current frame is truncated with an error indication set in the frame status word. The frame should subsequently be discarded by the user application.

41.3.7 Magic packet detection

Magic packet detection wakes a node that is put in power-down mode by the node management agent. Magic packet detection is supported only if the MAC is configured in sleep mode.

41.3.7.1 Sleep mode

To put the MAC in Sleep mode, set ENET*n*_ECR[SLEEP]. At the same time ENET*n*_ECR[MAGICEN] should also be set to enable magic packet detection.

In addition, if ENET is enabled, write 1 to ENET*n*_ECR[SLEEP] before entering into low power mode.

When the MAC is in Sleep mode:

- The transmit logic is disabled.
- The FIFO receive/transmit functions are disabled.
- The receive logic is kept in Normal mode, but it ignores all traffic from the line except magic packets. They are detected so that a remote agent can wake the node.

41.3.7.2 Magic packet detection

The core is designed to detect magic packets (See "Magic Packets" topics) with the destination address set to:

- Any multicast address
- The broadcast address
- The unicast address programmed in PADDR1/2

When a magic packet is detected, EIR[WAKEUP] is set and none of the statistic registers are incremented.

41.3.7.3 Wakeup

When a magic packet is detected, indicated by ENET*n*_EIR[WAKEUP], ENET*n*_ECR[SLEEP] should be cleared to resume normal operation of the MAC. Clearing the SLEEP bit automatically masks ENET*n*_ECR[MAGICEN], disabling magic packet detection.

41.3.8 IP accelerator functions

The following sections describe the IP accelerator functions.

41.3.8.1 Checksum calculation

The IP and ICMP, TCP, UDP checksums are calculated with one's complement arithmetic summing up 16-bit values.

- For ICMP, the checksum is calculated over the complete ICMP datagram, in other words without IP header.
- For TCP and UDP, the checksums contain the header and data sections and values from the IP header, which can be seen as a pseudo-header that is not actually present in the data stream.

Table 41-23. IPv4 pseudo-header for checksum calculation

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
Source address							
Destination address							
Zero		Protocol		TCP/UDP length			

Table 41-24. IPv6 pseudo-header for checksum calculation

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
Source address							
Destination address							
TCP/UDP length							
Zero						Next header	

The TCP/UDP length value is the length of the TCP or UDP datagram, which is equal to the payload of an IP datagram. It is derived by subtracting the IP header length from the complete IP datagram length that is given in the IP header (IPv4), or directly taken from the IP header (IPv6). The protocol field is the corresponding value from the IP header. The Zero fields are all zeroes.

For IPv6, the complete 128-bit addresses are considered. The next header value identifies the upper layer protocol as either TCP or UDP. It may differ from the next header value of the IPv6 header if extension headers are inserted before the protocol header.

The checksum calculation uses 16-bit words in network byte order: The first byte sent/received is the MSB, and the second byte sent/received is the LSB of the 16-bit value to add to the checksum. If the frame ends on an odd number of bytes, a zero byte is appended for checksum calculation only, and is not actually transmitted.

41.3.8.2 Additional padding processing

According to IEEE 802.3, any Ethernet frame must have a minimum length of 64 octets.

The MAC usually removes padding on receive when a frame with length information is received. Because IP frames have a type value instead of length, the MAC does not remove padding for short IP frames, as it is not aware of the frame contents.

The IP accelerator function can be configured to remove the Ethernet padding bytes that might follow the IP datagram.

On transmit, the MAC automatically adds padding as necessary to fill any frame to a 64-byte length.

41.3.8.3 32-bit Ethernet payload alignment

The data FIFOs allow inserting two additional arbitrary bytes in front of a frame. This extends the 14-byte Ethernet header to a 16-byte header, which leads to alignment of the Ethernet payload, following the Ethernet header, on a 32-bit boundary.

This function can be enabled for transmit and receive independently with the corresponding SHIFT16 bits in the ENET*n*_TACC and ENET*n*_RACC registers.

When enabled, the valid frame data is arranged as shown in Table 41-25.

Table 41-25. 64-bit interface data structure with SHIFT16 enabled

63 56	55 48	47 40	39 32	31 24	23 16	15 8	7 0

Table continues on the next page...

Table 41-25. 64-bit interface data structure with SHIFT16 enabled (continued)

Byte 5	Byte 4	Byte 3	Byte 2	Byte 1	Byte 0	Any value	Any value
Byte 13	Byte 12	Byte 11	Byte 10	Byte 9	Byte 8	Byte 7	Byte 6
...							

41.3.8.3.1 Receive processing

When ENET*n*_RACC[SHIFT16] is set, each frame is received with two additional bytes in front of the frame.

The user application must ignore these first two bytes and find the first byte of the frame in bits 23–16 of the first word from the RX FIFO.

Note

SHIFT16 must be set during initialization and kept set during the complete operation, because it influences the FIFO write behavior.

41.3.8.3.2 Transmit processing

When ENET*n*_TACC[SHIFT16] is set, the first two bytes of the first word written (bits 15–0) are discarded immediately by the FIFO write logic.

The SHIFT16 bit can be enabled/disabled for each frame individually if required, but can be changed only between frames.

41.3.8.4 Received frame discard

Because the receive FIFO must be operated in store and forward mode (ENET*n*_RSFL cleared), received frames can be discarded based on the following errors:

- The MAC function receives the frame with an error:
  - The frame has an invalid payload length
  - Frame length is greater than MAX_FL
  - Frame received with a CRC-32 error
  - Frame truncated due to receive FIFO overflow
  - Frame is corrupted as PHY signaled an error (RX_ERR asserted during reception)

- An IP frame is detected and the IP header checksum is wrong
- An IP frame with a valid IP header and a valid IP header checksum is detected, the protocol is known but the protocol-specific checksum is wrong

If one of the errors occurs and the IP accelerator function is configured to discard frames (ENET*n*_RACC), the frame is automatically discarded. Statistics are maintained normally and are not affected by this discard function.

41.3.8.5 IPv4 fragments

When an IPv4 IP fragment frame is received, only the IP header is inspected and its checksum verified. 32-bit alignment operates the same way on fragments as it does on normal IP frames, as specified above.

The IP fragment frame payload is not inspected for any protocol headers. As such, a protocol header would only exist in the very first fragment. To assist in protocol-specific checksum verification, the one's-complement sum is calculated on the IP payload (all bytes following the IP header) and provided with the frame status word.

The frame fragment status field (RxBD[FRAG]) is set to indicate a fragment reception, and the one's-complement sum of the IP payload is available in RxBD[Payload checksum].

Note

After all fragments have been received and reassembled, the application software can take advantage of the payload checksum delivered with the frame's status word to calculate the protocol-specific checksum of the datagram.

For example, if a TCP payload is delivered by multiple IP fragments, the application software can calculate the pseudo-header checksum value from the first fragment, and add the payload checksums delivered with the status for all fragments to verify the TCP datagram checksum.

41.3.8.6 IPv6 support

The following sections describe the IPv6 support.

41.3.8.6.1 Receive processing

An Ethernet frame of type 0x86DD identifies an IP Version 6 frame (IPv6) frame. If an IPv6 frame is received, the first IP header is inspected (first ten words), which is available in every IPv6 frame.

If the receive SHIFT16 function is enabled, the IP header is aligned on a 32-bit boundary allowing more efficient processing (see 32-bit Ethernet payload alignment).

For TCP and UDP datagrams, the pseudo-header checksum calculation is performed and verified.

To assist in protocol-specific checksum verification, the one's-complement sum is always calculated on the IP payload (all bytes following the IP header) and provided with the frame status word. For example, if extension headers were present, their sums can be subtracted in software from the checksum to isolate the TCP/UDP datagram checksum, if required.

41.3.8.6.2 Transmit processing

For IPv6 transmission, the SHIFT16 function is supported to process 32-bit aligned datagrams.

IPv6 has no IP header checksum; therefore, the IP checksum insertion configuration is ignored.

The protocol checksum is inserted only if the next header of the IP header is a known protocol (TCP, UDP, or ICMP). If a known protocol is detected, the checksum over all bytes following the IP header is calculated and inserted in the correct position.

The pseudo-header checksum calculation is performed for TCP and UDP datagrams accordingly.

41.3.9 Resets and stop controls

The following sections describe the resets and stop controls.

41.3.9.1 Hardware reset

To reset the Ethernet module, set ENET*n*_ECR[RESET].

41.3.9.2 Soft reset

When ENET*n*_ECR[ETHER_EN] is cleared during operation, the following occurs:

- uDMA, buffer descriptor, and FIFO control logic are reset, including the buffer descriptor and FIFO pointers.
- A currently ongoing transmit is terminated by asserting TXER to the PHY.
- A currently ongoing transmit FIFO write from the application is terminated by stopping the write to the FIFO, and all further data from the application is ignored. All subsequent writes are ignored until re-enabled.
- A currently ongoing receive FIFO read is terminated. The RxBD has arbitrary values in this case.

41.3.9.3 Debug mode

When the processor enters debug mode and ECR[DBGEN] is set, the MAC enters a freeze state where it stops all transmit and receive activities gracefully.

The following happens when the MAC enters hardware freeze:

- A currently ongoing receive transaction on the receive application interface is completed as normal. No further frames are read from the FIFO.
- A currently ongoing transmit transaction on the transmit application interface is completed as normal (in other words, until writing end-of-packet (EOP)).
- A currently ongoing frame receive is completed normally, after which no further frames are accepted from the MII.
- A currently ongoing frame transmit is completed normally, after which no further frames are transmitted.

41.3.9.4 Graceful stop

During a graceful stop, any currently ongoing transactions are completed normally and no further frames are accepted. The MAC can resume from a graceful stop without the need for a reset (for example, clearing ETHER_EN is not required).

The following conditions lead to a graceful stop of the MAC transmit or receive datapaths.

41.3.9.4.1 Graceful transmit stop (GTS)

When gracefully stopped, the MAC is no longer reading frame data from the transmit FIFO and has completed any ongoing transmission.

In any of the following conditions, the transmit datapath stops after an ongoing frame transmission has been completed normally.

- ENET*n*_TCR[GTS] is set by software.
- ENET*n*_TCR[TFC_PAUSE] is set by software requesting a pause frame transmission. The status (and register bit) is cleared after the pause frame has been sent.
- A pause frame was received stopping the transmitter. The stopped situation is terminated when the pause timer expires or a pause frame with zero quanta is received.
- MAC is placed in Sleep mode by software or the processor entering Stop mode (see Sleep mode).
- The MAC is in Hardware Freeze mode.

When the transmitter has reached its stopped state, the following events occur:

- The GRA interrupt is asserted, when transitioned into stopped state.
- In Hardware Freeze mode, the GRA interrupt does not wait for the application write completion and asserts when the transmit state machine (in other words, line side of TX FIFO) reaches its stopped state.

41.3.9.4.2 Graceful receive stop (GRS)

When gracefully stopped, the MAC is no longer writing frames into the receive FIFO.

The receive datapath stops after any ongoing frame reception has been completed normally, if any of the following conditions occur:

- MAC is placed in Sleep mode either by the software or the processor is in Stop mode). The MAC continues to receive frames and search for magic packets if enabled (see Magic packet detection). However, no frames are written into the receive FIFO, and therefore are not forwarded to the application.
- The MAC is in Hardware Freeze mode. The MAC does not accept any frames from the MII.

When the receive datapath is stopped, the following events occur:

- If the RX is in the stopped state, RCR[GRS] is set
- The GRA interrupt is asserted when the transmitter and receiver are stopped
- Any ongoing receive transaction to the application (RX FIFO read) continues normally until the frame end of package (EOP) is reached. After this, the following occurs:
  - When Sleep mode is active, all further frames are discarded, flushing the RX FIFO
  - In Hardware Freeze mode, no further frames are delivered to the application and they stay in the receive FIFO.

Note

The assertion of GRS does not wait for an ongoing FIFO read transaction on the application side of the FIFO (FIFO read).

41.3.9.4.3 Graceful stop interrupt (GRA)

The graceful stop interrupt (GRA) is asserted for the following conditions:

- In Sleep mode, the interrupt asserts only after both TX and RX datapaths are stopped.
- In Hardware Freeze mode, the interrupt asserts only after both TX and RX datapaths are stopped.
- The MAC transmit datapath is stopped for any other condition (GTS, TFC_PAUSE, pause received).

The GRA interrupt is triggered only once when the stopped state is entered. If the interrupt is cleared while the stop condition persists, no further interrupt is triggered.

41.3.10 IEEE 1588 functions

To allow for IEEE 1588 or similar time synchronization protocol implementations, the MAC is combined with a time-stamping module to support precise time-stamping of incoming and outgoing frames. Set ENET*n*_ECR[EN1588] to enable 1588 support.

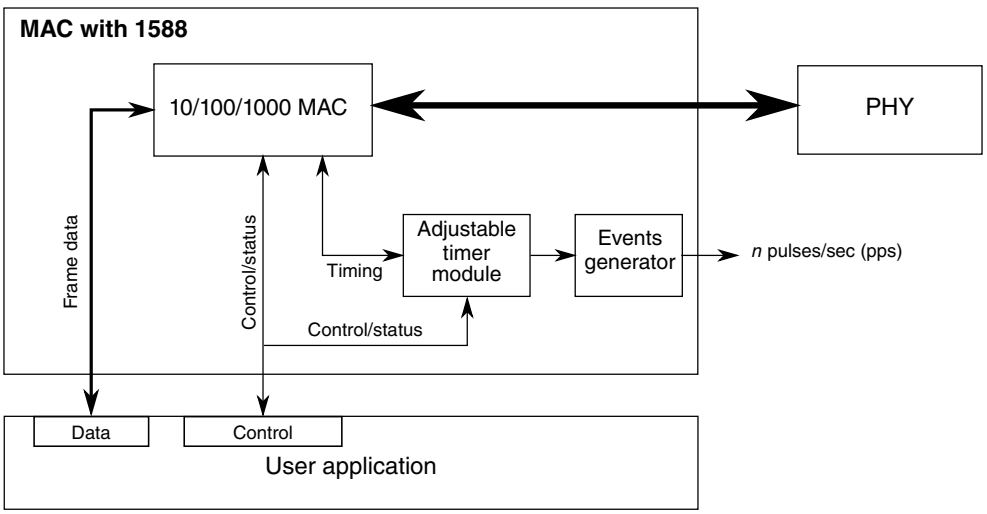


Figure 41-9. IEEE 1588 functions overview

41.3.10.1 Adjustable timer module

The adjustable timer module (TSM) implements the free-running counter (FRC), which generates the timestamps. The FRC operates with the time-stamping clock, which can be set to any value depending on your system requirements.

Through dedicated correction logic, the timer can be adjusted to allow synchronization to a remote master and provide a synchronized timing reference to the local system. The timer can be configured to cause an interrupt after a fixed time period, to allow synchronization of software timers or perform other synchronized system functions.

The timer is typically used to implement a period of one second; hence, its value ranges from 0 to (1×10^9)-1. The period event can trigger an interrupt, and software can maintain the seconds and hours time values as necessary.

41.3.10.1.1 Adjustable timer implementation

The adjustable timer consists of a programmable counter/accumulator and a correction counter. The periods of both counters and their increment rates are freely configurable, allowing very fine tuning of the timer.

See Timer Synchronization for Multi-Port Implementations, for external clock input options.

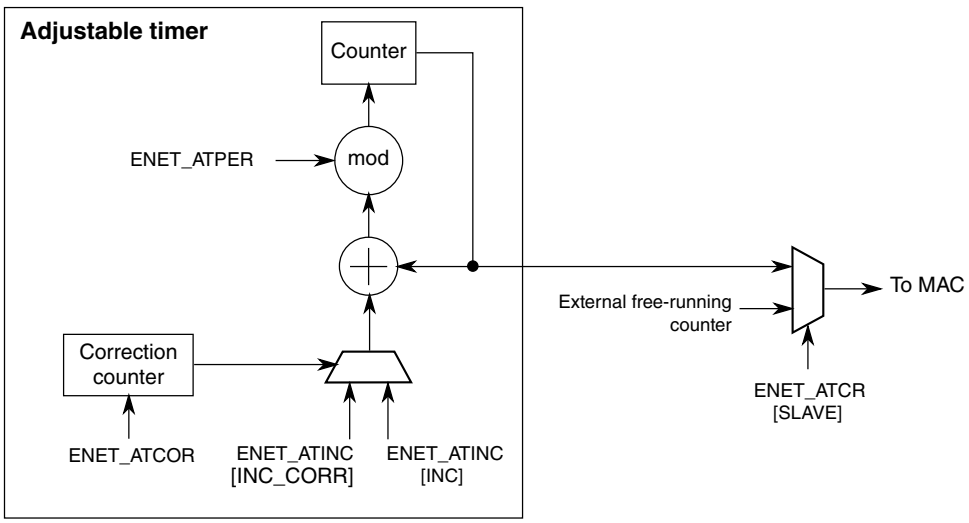


Figure 41-10. Adjustable timer implementation detail

The counter produces the current time. During each time-stamping clock cycle, a constant value is added to the current time as programmed in ENET*n*_ATINC. The value depends on the chosen time-stamping clock frequency. For example, if it operates at 125 MHz, setting the increment to eight represents 8 ns.

The period, configured in ENET*n*_ATPER, defines the modulo when the counter wraps. In a typical implementation, the period is set to 1×10^9 so that the counter wraps every second, and hence all timestamps represent the absolute nanoseconds within the one second period. When the period is reached, the counter wraps to start again respecting the period modulo. This means it does not necessarily start from zero, but instead the counter is loaded with the value (Current + Inc $-(1 \times 10^9)$), assuming the period is set to 1×10^9.

The correction counter operates fully independently, and increments by one with each time-stamping clock cycle. When it reaches the value configured in ENET*n*_ATCOR, it restarts and instructs the timer once to increment by the correction value, instead of the normal value.

The normal and correction increments are configured in ENET*n*_ATINC. To speed up the timer, set the correction increment more than the normal increment value. To slow down the timer, set the correction increment less than the normal increment value.

The correction counter only defines the distance of the corrective actions, not the amount. This allows very fine corrections and low jitter (in the range of 1 ns) independent of the chosen clock frequency.

By enabling slave mode (ENET*n*_ATCR[SLAVE] = 1), the timer is ignored and the current time is externally provided from one of the external modules. See the Chip Configuration details for which clock source is used. This is useful if multiple modules within the system must operate from a single timer (see Timer Synchronization for Multi-Port Implementations). When slave mode is enabled, you still must set ENET*n*_ATINC[INC] to the value of the master, since it is used for internal comparisons.

41.3.10.2 Timer Synchronization for Multi-Port Implementations

Additional inputs are available to provide a timer value for all time-stamping functions. This is necessary to synchronize the two MACs to a single reference timer. See the Chip Configuration details for available clock inputs to the time-stampimg functions. To operate the MAC in slave mode, ENET*n*_ATCR[SLAVE] disables the internal adjustable timer and uses the externally provided timer.

41.3.10.3 Transmit timestamping

Only 1588 event frames need to be time-stamped on transmit. The client application (for example, the MAC driver) should detect 1588 event frames and set TxBD[TS] together with the frame.

If TxBD[TS] is set, the MAC records the timestamp for the frame in ENET*n*_ATSTMP. ENET*n*_EIR[TS_AVAIL] is set to indicate that a new timestamp is available.

Software implements a handshaking procedure by setting TxBD[TS] when it transmits the frame for which a timestamp is needed, and then waits for ENET*n*_EIR[TS_AVAIL] to determine when the timestamp is available. The timestamp is then read from ENET*n*_ATSTMP. This is done for all event frames. Other frames do not use TxBD[TS] and, therefore, do not interfere with the timestamp capture.

41.3.10.4 Receive timestamping

When a frame is received, the MAC latches the value of the timer when the frame's start of frame delimiter (SFD) field is detected, and provides the captured timestamp on RxBD[1588 timestamp]. This is done for all received frames.

41.3.10.5 Time synchronization

The adjustable timer module is available to synchronize the local clock of a node to a remote master. It implements a free running 32-bit counter, and also contains an additional correction counter.

The correction counter increases or decreases the rate of the free running counter, enabling very fine granular changes of the timer for synchronization, yet adding only very low jitter when performing corrections.

The application software implements, in a slave scenario, the required control algorithm, setting the correction to compensate for local oscillator drifts and locking the timer to the remote master clock on the network.

The timer and all timestamp-related information should be configured to show the true nanoseconds value of a second (in other words, the timer is configured to have a period of one second). Hence, the values range from 0 to (1×10^9)–1. In this application, the seconds counter is implemented in software using an interrupt function that is executed when the nanoseconds counter wraps at 1×10^9.

41.3.10.6 Input Capture and Output Compare

The Input Capture Output Compare block can be used to provide precise hardware timing for input and output events.

41.3.10.6.1 Input capture

The TCCR*n* capture registers latch the time value when the corresponding external event occurs. An event can be a rising-, falling-, or either-edge of one of the 1588_TMR*n* signals. An event will cause the corresponding TCSR*n*[TF] timer flag to be set, indicating that an input capture has occurred. If the corresponding interrupt is enabled with the TCSR*n*[TIE] field, an interrupt can be generated.

41.3.10.6.2 Output compare

The TCCR*n* compare registers are loaded with the time at which the corresponding event should occur. When the ENET free-running counter value matches the output compare reference value in the TCCR*n* register, the corresponding flag, TCSR*n*[TF], is set, indicating that an output compare has occurred. The corresponding interrupt, if enabled by TCSR*n*[TIE], will be generated.The corresponding 1588_TMR*n* output signal will be asserted according to TCSR*n*[TMODE].

NOTE

If TSCR*n*[TMODE] is set to 10X1b or 1010b then the timer output pin toggle-on-overflow will occur only when PINPER, PEREN, and EN bits of the ATCR register are one.

41.3.10.6.3 DMA requests

A DMA request can be enabled by setting TCSR*n*[TDRE]. The corresponding DMA request is generated when the TCSR*n*[TF] timer flag is set. When the DMA has completed, the corresponding TCSRn[TF] flag is cleared.

41.3.11 FIFO thresholds

The core FIFO thresholds are fully programmable to dynamically change the FIFO operation.

For example, store and forward transfer can be enabled by a simple change in the FIFO threshold registers.

The thresholds are defined in 64-bit words.

The receive and transmit FIFOs both have a depth of 256 words.

41.3.11.1 Receive FIFO

Four programmable thresholds are available, which can be set to any value to control the core operation as follows.

Table 41-26. Receive FIFO thresholds definition

Register	Description
ENET*n*_RSFL [RX_SECTION_FULL]	When the FIFO level reaches the ENET*n*_RSFL value, the MAC status signal is asserted to indicate that data is available in the receive FIFO (cut-through operation). Once asserted, if the FIFO empties below the threshold set with ENET*n*_RAEM and if the end-of-frame is not yet stored in the FIFO, the status signal is deasserted again. If a frame has a size smaller than the threshold (in other words, an end-of-frame is available for the frame), the status is also asserted. To enable store and forward on the receive path, clear ENET*n*_RSFL. The MAC status signal is asserted only when a complete frame is stored in the receive FIFO. When programming a non-zero value to ENET*n*_RSFL (cut-through operation) it should be greater than ENET*n*_RAEM.
ENET*n*_RAEM	When the FIFO level reaches the ENET*n*_RAEM value and the end-of-frame has not been received, the core receive read control stops the FIFO read (and subsequently stops transferring data to the MAC client application).

Table continues on the next page...

Table 41-26. Receive FIFO thresholds definition (continued)

Register	Description
[RX_ALMOST_EMPTY]	It continues to deliver the frame, if again more data than the threshold or the end-of-frame is available in the FIFO. Set ENET*n*_RAEM to a minimum of six.
ENET*n*_RAFL [RX_ALMOST_FULL]	When the FIFO level approaches the maximum and there is no more space remaining for at least ENETn_RAFL number of words, the MAC control logic stops writing data in the FIFO and truncates the receive frame to avoid FIFO overflow. The corresponding error status is set when the frame is delivered to the application. Set ENET*n*_RAFL to a minimum of 4.
ENET*n*_RSEM [RX_SECTION_EMPTY]	When the FIFO level reaches the ENET*n*_RSEM value, an indication is sent to the MAC transmit logic, which generates an XOFF pause frame. This indicates FIFO congestion to the remote Ethernet client. When the FIFO level goes below the value programmed in ENET*n*_RSEM, an indication is sent to the MAC transmit logic, which generates an XON pause frame. This indicates the FIFO congestion is cleared to the remote Ethernet client. Clearing ENET*n*_RSEM disables any pause frame generation.

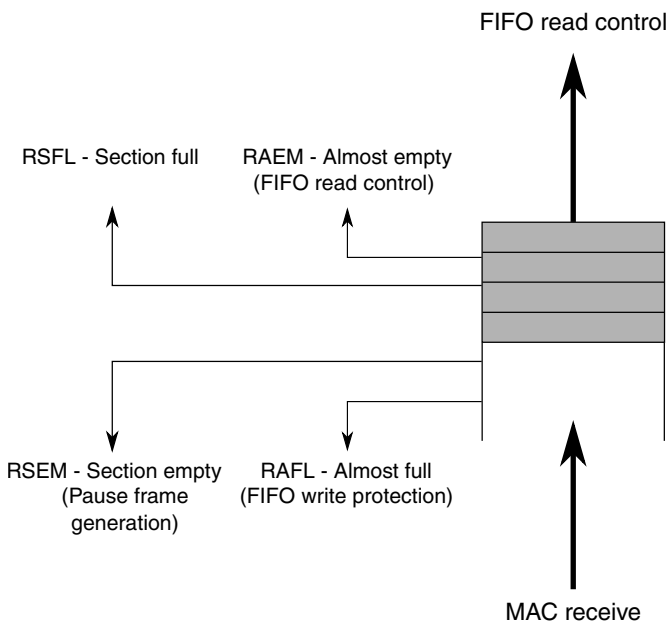


Figure 41-11. Receive FIFO overview

41.3.11.2 Transmit FIFO

Four programmable thresholds are available which control the core operation as described below.

Table 41-27. Transmit FIFO thresholds definition

Register	Description
ENET*n*_TAEM [TX_ALMOST _EMPTY]	When the FIFO level reaches the ENET*n*_TAEM value and no end-of-frame is available for the frame, the MAC transmit logic avoids a FIFO underflow by stopping FIFO reads and transmitting the Ethernet frame with an MII error indication. Set ENET*n*_TAEM to a minimum of 4.
ENET*n*_TAFL [TX_ALMOST _FULL]	When the FIFO level approaches the maximum, so that there is no more space for at least ENET*n*_TAFL number of words, the MAC deasserts its control signal to the application. If the application does not react on this signal, the FIFO write control logic avoids FIFO overflow by truncating the current frame and setting the error status. As a result, the frame is transmitted with an MII error indication. Set ENET*n*_TAFL to a minimum of 4. Larger values allow more latency for the application to react on the MAC control signal deassertion, before the frame is truncated. A typical setting is 8, which offers 3–4 clock cycles of latency to the application to react on the MAC control signal deassertion.
ENET*n*_TSEM [TX_SECTION _EMPTY]	When the FIFO level reaches the ENET*n*_TSEM value, a MAC status signal is deasserted to indicate that the transmit FIFO is getting full. This gives the ENET module an indication to slow or stop its write transaction to avoid a buffer overflow. This is a pure indication function to the application. It has no effect within the MAC. When ENET*n*_TSEM is 0, the signal is never deasserted.
ENET*n*_TFWR	When the FIFO level reaches the ENET*n*_TFWR value and when STRFWD is cleared, the MAC transmit control logic starts frame transmission before the end-of-frame is available in the FIFO (cut-through operation). If a complete frame has a size smaller than the ENET*n*_TFWR threshold, the MAC also transmits the frame to the line. To enable store and forward on the transmit path, set STRFWD. In this case, the MAC starts to transmit data only when a complete frame is stored in the transmit FIFO.

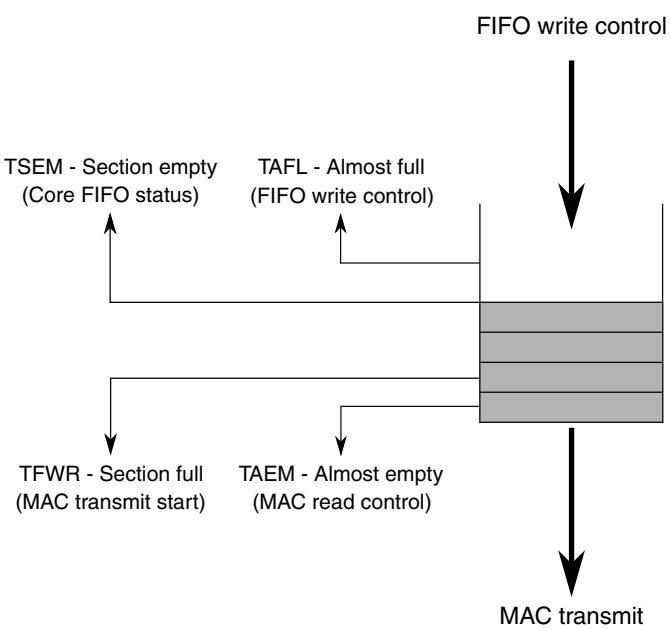


Figure 41-12. Transmit FIFO overview

41.3.12 Loopback options

The core implements internal loopback options, which are controlled by the following ENET*n*_RCR register fields:

Table 41-28. Loopback options

Register field	Description
LOOP	Internal MII loopback. The MAC transmit is returned to the MAC receive. No data is transmitted to the external interfaces. In MII internal loopback, MII_TXCLK and MII_RXCLK must be provided with a clock signal (2.5 MHz for 10 Mbit/s, and 25 MHz for 100 Mbit/s))

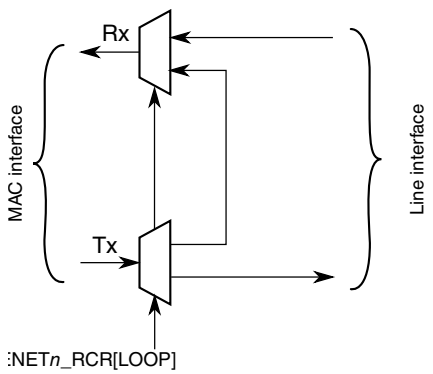


Figure 41-13. Loopback options

41.3.13 Legacy buffer descriptors

To support the Ethernet controller on previous chips, legacy FEC buffer descriptors are available. To enable legacy support, write 0 to ENET*n*_ECR[1588EN].

NOTE

- The legacy buffer descriptor tables show the byte order for little-endian chips. ECR[DBSWP] must be set to 1 after reset to enable little-endian mode.

41.3.13.1 Legacy receive buffer descriptor

The following table shows the legacy FEC receive buffer descriptor. Table 41-32 contains the descriptions for each field.

Table 41-29. Legacy FEC receive buffer descriptor (RxBD)

	Byte 1								Byte 0							
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Offset + 0	Data length															
Offset + 2	E	RO1	W	RO2	L	—	—	M	BC	MC	LG	NO	—	CR	OV	TR
Offset + 4	Rx data buffer pointer — low halfword															
Offset + 6	Rx data buffer pointer — high halfword															

41.3.13.2 Legacy transmit buffer descriptor

The following table shows the legacy FEC transmit buffer descriptor. Table 41-34 contains the descriptions for each field.

Table 41-30. Legacy FEC transmit buffer descriptor (TxBD)

	Byte 1								Byte 0							
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Offset + 0	Data Length															
Offset + 2	R	TO1	W	TO2	L	TC	ABC[1]	—	—	—	—	—	—	—	—	—
Offset + 4	Tx Data Buffer Pointer — low halfword															
Offset + 6	Tx Data Buffer Pointer — high halfword															

1. This field is not supported by the uDMA.

41.3.14 Enhanced buffer descriptors

This section provides a description of the enhanced operation of the driver/uDMA via the buffer descriptors.

It is followed by a detailed description of the receive and transmit descriptor fields. To enable the enhanced features, set ENET*n*_ECR[1588EN].

NOTE

The enhanced buffer descriptor tables show the byte order for little-endian chips. ECR[DBSWP] must be set to 1 after reset to enable little-endian mode.

41.3.14.1 Enhanced receive buffer descriptor

The following table shows the enhanced uDMA receive buffer descriptor. Table 41-32 contains the descriptions for each field.

Table 41-31. Enhanced uDMA receive buffer descriptor (RxBD)

	Byte 1								Byte 0							
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Offset + 0	Data length															
Offset + 2	E	RO1	W	RO2	L	—	—	M	BC	MC	LG	NO	—	CR	OV	TR
Offset + 4	Rx data buffer pointer – low halfword															
Offset + 6	Rx data buffer pointer – high halfword															
Offset + 8	VPCP			—	—	—	—	—	—	—	ICE	PCR	—	VLAN	IPV6	FRAG
Offset + A	ME	—	—	—	—	PE	CE	UC	INT	—	—	—	—	—	—	—
Offset + C	Payload checksum															
Offset + E	Header length					—	—	—	Protocol type							
Offset + 10	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Offset + 12	BDU	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Offset + 14	1588 timestamp – low halfword															
Offset + 16	1588 timestamp – high halfword															
Offset + 18	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Offset + 1A	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Offset + 1C	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Offset + 1E	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Table 41-32. Receive buffer descriptor field definitions

Word	Field	Description
Offset + 0	15–0 Data Length	Data length. Written by the MAC. Data length is the number of octets written by the MAC into this BD's data buffer if L is cleared (the value is equal to EMRBR), or the length of the frame including CRC if L is set. It is written by the MAC once as the BD is closed.
Offset + 2	15 E	Empty. Written by the MAC (= 0) and user (= 1). 0 The data buffer associated with this BD is filled with received data, or data reception has aborted due to an error condition. The status and length fields have been updated as required. 1 The data buffer associated with this BD is empty, or reception is currently in progress.
Offset + 2	14 RO1	Receive software ownership. This field is reserved for use by software. This read/write field is not modified by hardware, nor does its value affect hardware.
Offset + 2	13 W	Wrap. Written by user. 0 The next buffer descriptor is found in the consecutive location. 1 The next buffer descriptor is found at the location defined in ENET*n*_RDSR.
Offset + 2	12	Receive software ownership. This field is reserved for use by software. This read/write field is not modified by hardware, nor does its value affect hardware.

Table continues on the next page...

Table 41-32. Receive buffer descriptor field definitions (continued)

Word	Field	Description
	RO2	
Offset + 2	11 L	Last in frame. Written by the uDMA. 0 The buffer is not the last in a frame. 1 The buffer is the last in a frame.
Offset + 2	10–9	Reserved, must be cleared.
Offset + 2	8 M	Miss. Written by the MAC. This field is set by the MAC for frames accepted in promiscuous mode, but flagged as a miss by the internal address recognition. Therefore, while in promiscuous mode, you can use the this field to quickly determine whether the frame was destined to this station. This field is valid only if the L and PROM bits are set. 0 The frame was received because of an address recognition hit. 1 The frame was received because of promiscuous mode. The information needed for this field comes from the promiscuous_miss(ff_rx_err_stat[26]) sideband signal.
Offset + 2	7 BC	Set if the DA is broadcast (FFFF_FFFF_FFFF).
Offset + 2	6 MC	Set if the DA is multicast and not BC.
Offset + 2	5 LG	Receive frame length violation. Written by the MAC. A frame length greater than RCR[MAX_FL] was recognized. This field is valid only if the L field is set. The receive data is not altered in any way unless the length exceeds TRUNC_FL bytes.
Offset + 2	4 NO	Receive non-octet aligned frame. Written by the MAC. A frame that contained a number of bits not divisible by 8 was received, and the CRC check that occurred at the preceding byte boundary generated an error or a PHY error occurred. This field is valid only if the L field is set. If this field is set, the CR field is not set.
Offset + 2	3	Reserved, must be cleared.
Offset + 2	2 CR	Receive CRC or frame error. Written by the MAC. This frame contains a PHY or CRC error and is an integral number of octets in length. This field is valid only if the L field is set.
Offset + 2	1 OV	Overrun. Written by the MAC. A receive FIFO overrun occurred during frame reception. If this field is set, the other status fields, M, LG, NO, and CR, lose their normal meaning and are zero. This field is valid only if the L field is set.
Offset + 2	0 TR	Set if the receive frame is truncated (frame length >TRUNC_FL). If the TR field is set, the frame must be discarded and the other error fields must be ignored because they may be incorrect.
Offset + 4	15–0 Data buffer pointer low	Receive data buffer pointer, low halfword
0ffset + 6	15–0 Data buffer pointer high	Receive data buffer pointer, high halfword[1]
Offset + 8	15–13 VPCP	VLAN priority code point. This field is written by the uDMA to indicate the frame priority level. Valid values are from 0 (best effort) to 7 (highest). This value can be used to prioritize different classes of traffic (e.g., voice, video, data). This field is only valid if the L field is set.
Offset + 8	12–6	Reserved, must be cleared.

Table continues on the next page...

Table 41-32. Receive buffer descriptor field definitions (continued)

Word	Field	Description
Offset + 8	5 ICE	IP header checksum error. This is an accelerator option. This field is written by the uDMA. Set when either a non-IP frame is received or the IP header checksum was invalid. An IP frame with less than 3 bytes of payload is considered to be an invalid IP frame. This field is only valid if the L field is set.
Offset + 8	4 PCR	Protocol checksum error. This is an accelerator option. This field is written by the uDMA. Set when the checksum of the protocol is invalid or an unknown protocol is found and checksumming could not be performed. This field is only valid if the L field is set.
Offset + 8	3	Reserved, must be cleared.
Offset + 8	2 VLAN	VLAN. This is an accelerator option. This field is written by the uDMA. It means that the frame has a VLAN tag. This field is valid only if the L field is set.
Offset + 8	1 IPV6	IPV6 Frame. This field is written by the uDMA. This field indicates that the frame has an IPv6 frame type. If this field is not set it means that an IPv4 or other protocol frame was received. This field is valid only if the L field is set.
Offset + 8	0 FRAG	IPv4 Fragment.This is an accelerator option.This field is written by the uDMA. It indicates that the frame is an IPv4 fragment frame. This field is only valid when the L field is set.
Offset + A	15 ME	MAC error. This field is written by the uDMA. This field means that the frame stored in the system memory was received with an error (typically, a receive FIFO overflow). This field is only valid when the L field is set.
Offset + A	14–11	Reserved, must be cleared.
Offset + A	10 PE	PHY Error. This field is written by the uDMA. Set to "1"when the frame was received with an Error character on the PHY interface. The frame is invalid. This field is valid only when the L field is set.
Offset + A	9 CE	Collision. This field is written by the uDMA. Set when the frame was received with a collision detected during reception. The frame is invalid and sent to the user application. This field is valid only when the L field is set.
Offset + A	8 UC	Unicast. This field is written by the uDMA, and means that the frame is unicast. This field is valid regardless of whether the L field is set.
Offset + A	7 INT	Generate RXB/RXF interrupt. This field is set by the user to indicate that the uDMA is to generate an interrupt on the *dma_int_rxb* / *dma_int_rxfevent*.
Offset + A	6–0	Reserved, must be cleared.
Offset + C	15–0 Payload checksum	Internet payload checksum. This is an accelerator option. It is the one's complement sum of the payload section of the IP frame. The sum is calculated over all data following the IP header until the end of the IP payload. This field is valid only when the L field is set.
Offset + E	15–11 Header length	Header length. This is an accelerator option. This field is written by the uDMA. This field is the sum of 32-bit words found within the IP and its following protocol headers. If an IP datagram with an unknown protocol is found, then the value is the length of the IP header. If no IP frame or an erroneous IP header is found, the value is 0. The following values are minimum values if no header options exist in the respective headers: • ICMP/IP: 6 (5 IP header, 1 ICMP header) • UDP/IP: 7 (5 IP header, 2 UDP header) • TCP/IP: 10 (5 IP header, 5 TCP header) This field is only valid if the L field is set.
Offset + E	10–8	Reserved, must be cleared.

Table continues on the next page...

Table 41-32. Receive buffer descriptor field definitions (continued)

Word	Field	Description
Offset + E	7–0 Protocol type	Protocol type. This is an accelerator option. The 8-bit protocol field found within the IP header of the frame. It is valid only when ICE is cleared. This field is valid only when the L field is set.
Offset + 10	15–0	Reserved, must be cleared.
Offset + 12	15 BDU	Last buffer descriptor update done. Indicates that the last BD data has been updated by uDMA. This field is written by the user (=0) and uDMA (=1).
Offset + 12	14–0	Reserved, must be cleared.
Offset + 14	15–0	This value is written by the uDMA. It is only valid if the L field is set.
Offset + 16	1588 timestamp	
Offset + 18 – Offset + 1E	15–0	Reserved, must be cleared.

1. The receive buffer pointer, containing the address of the associated data buffer, must always be evenly divisible by 64. The buffer must reside in memory external to the MAC. The Ethernet controller never modifies this value.

41.3.14.2 Enhanced transmit buffer descriptor

Table 41-33. Enhanced transmit buffer descriptor (TxBD)

	Byte 1								Byte 0							
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Offset + 0	Data length															
Offset + 2	R	TO1	W	TO2	L	TC	—	—	—	—	—	—	—	—	—	—
Offset + 4	Tx Data Buffer Pointer – low halfword															
Offset + 6	Tx Data Buffer Pointer – high halfword															
Offset + 8	TXE	—	UE	EE	FE	LCE	OE	TSE	—	—	—	—	—	—	—	—
Offset + A	—	INT	TS	PINS	IINS	—	—	—	—	—	—	—	—	—	—	—
Offset + C	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Offset + E	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Offset + 10	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Offset + 12	BDU	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Offset + 14	1588 timestamp – low halfword															
Offset + 16	1588 timestamp – high halfword															
Offset + 18	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Offset + 1A	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Offset + 1C	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Offset + 1E	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Table 41-34. Enhanced transmit buffer descriptor field definitions

Word	Field	Description
Offset + 0	15–0 Data Length	Data length, written by user. Data length is the number of octets the MAC should transmit from this BD's data buffer. It is never modified by the MAC.
Offset + 2	15 R	Ready. Written by the MAC and you. 0 The data buffer associated with this BD is not ready for transmission. You are free to manipulate this BD or its associated data buffer. The MAC clears this field after the buffer has been transmitted or after an error condition is encountered. 1 The data buffer, prepared for transmission by you, has not been transmitted or currently transmits. You may write no fields of this BD after this field is set.
Offset + 2	14 TO1	Transmit software ownership. This field is reserved for software use. This read/write field is not modified by hardware and its value does not affect hardware.
Offset + 2	13 W	Wrap. Written by user. 0 The next buffer descriptor is found in the consecutive location 1 The next buffer descriptor is found at the location defined in ETDSR.
Offset + 2	12 TO2	Transmit software ownership. This field is reserved for use by software. This read/write field is not modified by hardware and its value does not affect hardware.
Offset + 2	11 L	Last in frame. Written by user. 0 The buffer is not the last in the transmit frame 1 The buffer is the last in the transmit frame
Offset + 2	10 TC	Transmit CRC. Written by user, and valid only when L is set. 0 End transmission immediately after the last data byte 1 Transmit the CRC sequence after the last data byte This field is valid only when the L field is set.
Offset + 2	9 ABC	Append bad CRC. **Note:** This field is not supported by the uDMA and is ignored.
Offset + 2	8–0	Reserved, must be cleared.
Offset + 4	15–0 Data buffer pointer low	Tx data buffer pointer, low halfword
Offset + 6	15–0 Data buffer pointer high	Tx data buffer pointer, high halfword. The buffer must reside in memory external to the MAC. This value is never modified by the Ethernet controller. **NOTE:** For optimal performance, make the transmit buffer pointer evenly divisible by 64.
Offset + 8	15 TXE	Transmit error occurred. This field is written by the uDMA. This field indicates that there was a transmit error of some sort reported with the frame. Effectively this field is an OR of the other error fields including UE, EE, FE, LCE, OE, and TSE. This field is valid only when the L field is set.
Offset + 8	14	Reserved, must be cleared.
Offset + 8	13 UE	Underflow error. This field is written by the uDMA. This field indicates that the MAC reported an underflow error on transmit. This field is valid only when the L field is set.

Table continues on the next page...

Table 41-34. Enhanced transmit buffer descriptor field definitions (continued)

Word	Field	Description
Offset + 8	12 EE	Excess Collision error. This field is written by the uDMA. This field indicates that the MAC reported an excess collision error on transmit. This field is valid only when the L field is set.
Offset + 8	11 FE	Frame with error. This field is written by the uDMA. This field indicates that the MAC reported that the uDMA reported an error when providing the packet. This field is valid only when the L field is set.
Offset + 8	10 LCE	Late collision error. This field is written by the uDMA. This field indicates that the MAC reported that there was a Late Collision on transmit. This field is valid only when the L field is set.
Offset + 8	9 OE	Overflow error. This field is written by the uDMA. This field indicates that the MAC reported that there was a FIFO overflow condition on transmit. This field is only valid when the L field is set.
Offset + 8	8 TSE	Timestamp error. This field is written by the uDMA. This field indicates that the MAC reported a different frame type then a timestamp frame. This field is valid only when the L field is set.
Offset + 8	7–0	Reserved, must be cleared.
Offset + A	15	Reserved, must be cleared.
Offset + A	14 INT	Generate interrupt flags. This field is written by the user. This field is valid regardless of the L field and must be the same for all EBD for a given frame. The uDMA does not update this value.
Offset + A	13 TS	Timestamp. This field is written by the user. This indicates that the uDMA is to generate a timestamp frame to the MAC. This field is valid regardless of the L field and must be the same for all EBD for the given frame. The uDMA does not update this value.
Offset + A	12 PINS	Insert protocol specific checksum. This field is written by the user. If set, the MAC's IP accelerator calculates the protocol checksum and overwrites the corresponding checksum field with the calculated value. The checksum field must be cleared by the application generating the frame. The uDMA does not update this value. This field is valid regardless of the L field and must be the same for all EBD for a given frame.
Offset + A	11 IINS	Insert IP header checksum. This field is written by the user. If set, the MAC's IP accelerator calculates the IP header checksum and overwrites the corresponding header field with the calculated value. The checksum field must be cleared by the application generating the frame. The uDMA does not update this value. This field is valid regardless of the L field and must be the same for all EBD for a given frame.
Offset + A	10–0	Reserved, must be cleared.
Offset + C	15–0	Reserved, must be cleared.
Offset + E	15–0	Reserved, must be cleared.
Offset + 10	15–0	Reserved, must be cleared.
Offset + 12	15 BDU	Last buffer descriptor update done. Indicates that the last BD data has been updated by uDMA. This field is written by the user (=0) and uDMA (=1).
Offset + 12	14–0	Reserved, must be cleared.
Offset + 14	15–0	This value is written by the uDMA . It is valid only when the L field is set.
Offset + 16	1588 timestamp	
Offset + 18–Offset + 1E	15–0	Reserved, must be cleared.

41.3.15 Client FIFO application interface

The FIFO interface is completely asynchronous from the Ethernet line, and the transmit and receive interface can operate at a different clock rate.

All transfers to/from the user application are handled independently of the core operation, and the core provides a simple interface to user applications based on a two-signal handshake.

41.3.15.1 Data structure description

The data structure defined in the following tables for the FIFO interface must be respected to ensure proper data transmission on the Ethernet line. Byte 0 is sent to and received from the line first.

Table 41-35. FIFO interface data structure

	63 56	55 48	47 40	39 32	31 24	23 16	15 8	7 0
Word 0	Byte 7	Byte 6	Byte 5	Byte 4	Byte 3	Byte 2	Byte 1	Byte 0
Word 1	Byte 15	Byte 14	Byte 13	Byte 12	Byte 11	Byte 10	Byte 9	Byte 8
...	...							

The size of a frame on the FIFO interface may not be a modulo of 64-bit.

The user application may not care about the Ethernet frame formats in full detail. It needs to provide and receive an Ethernet frame with the following structure:

- Ethernet MAC destination address
- Ethernet MAC source address
- Optional 802.1q VLAN tag (VLAN type and info field)
- Ethernet length/type field
- Payload

Frames on the FIFO interface do not contain preamble and SFD fields, which are inserted and discarded by the MAC on transmit and receive, respectively.

- On receive, CRC and frame padding can be stripped or passed through transparently.
- On transmit, padding and CRC can be provided by the user application, or appended automatically by the MAC independently for each frame. No size restrictions apply.

Note

On transmit, if ENET*n*_TCR[ADDINS] is set, bytes 6–11 of each frame can be set to any value, since the MAC overwrites the bytes with the MAC address programmed in the ENET*n*_PAUR and ENET*n*_PALR registers.

Table 41-36. FIFO interface frame format

Byte number	Field
0–5	Destination MAC address
6–11	Source MAC address
12–13	Length/type field
14–N	Payload data

VLAN-tagged frames are supported on both transmit and receive, and implement additional information (VLAN type and info).

Table 41-37. FIFO interface VLAN frame format

Byte number	Field
0–5	Destination MAC address
6–11	Source MAC address
12–15	VLAN tag and info
16–17	Length/type field
18–N	Payload data

Note

The standard defines that the LSB of the MAC address is sent/received first, while for all the other header fields — in other words, length/type, VLAN tag, VLAN info, and pause quanta — the MSB is sent/received first.

41.3.15.2 Data structure examples

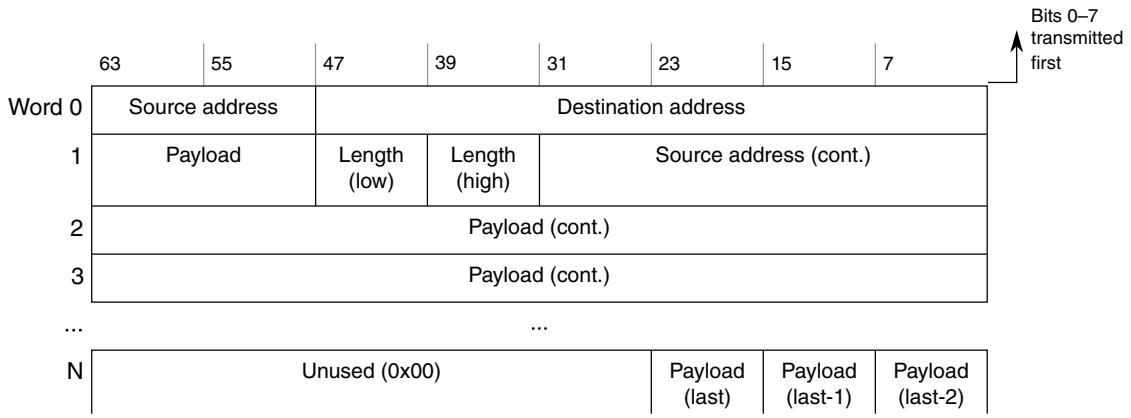


Figure 41-14. Normal Ethernet frame 64-bit mapping example

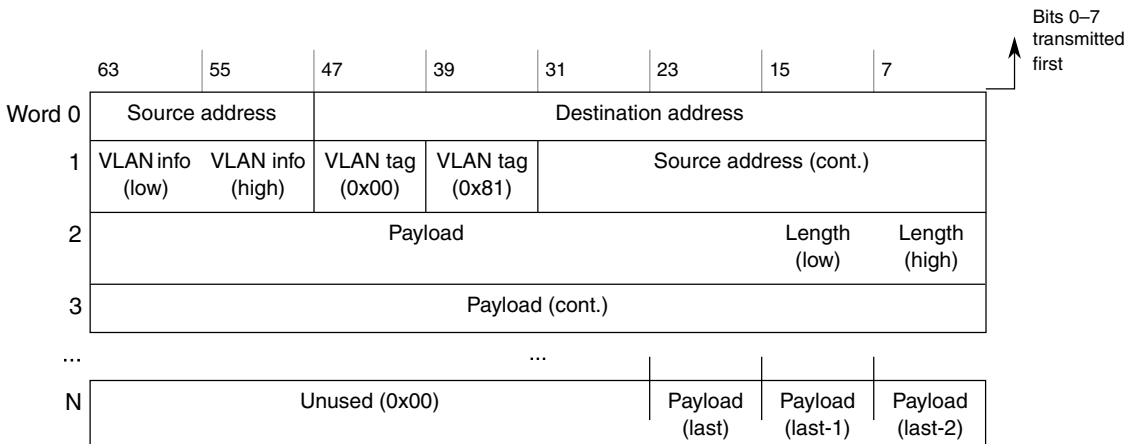


Figure 41-15. VLAN-tagged frame 64-bit mapping example

If CRC forwarding is enabled (CRCFWD = 0), the last four valid octets of the frame contain the FCS field. The non-significant bytes of the last word can have any value.

41.3.15.3 Frame status

A MAC layer status word and an accelerator status word is available in the receive buffer descriptor.

See Enhanced buffer descriptors for details.

The status is available with each frame with the last data of the frame.

If the frame status contains a MAC layer error (for example, CRC or length error), RxBD[ME] is also set with the last data of the frame.

41.3.16 FIFO protection

The following sections describe the FIFO protection mechanisms.

41.3.16.1 Transmit FIFO underflow

During a frame transfer, when the transmit FIFO reaches the almost empty threshold with no end-of-frame indication stored in the FIFO, the MAC logic:

- Stops reading data from the FIFO
- Asserts the MII error signal (MII_TXER) (1 in Figure 41-16) to indicate that the fragment already transferred is not valid
- Deasserts the MII transmit enable signal (MII_TXEN) to terminate the frame transfer (2)

After an underflow, when the application completes the frame transfer (3), the MAC transmit logic discards any new data available in the FIFO until the end of packet is reached (4) and sets the enhanced TxBD[UE] field.

The MAC starts to transfer data on the MII interface when the application sends a new frame with a start of frame indication (5).

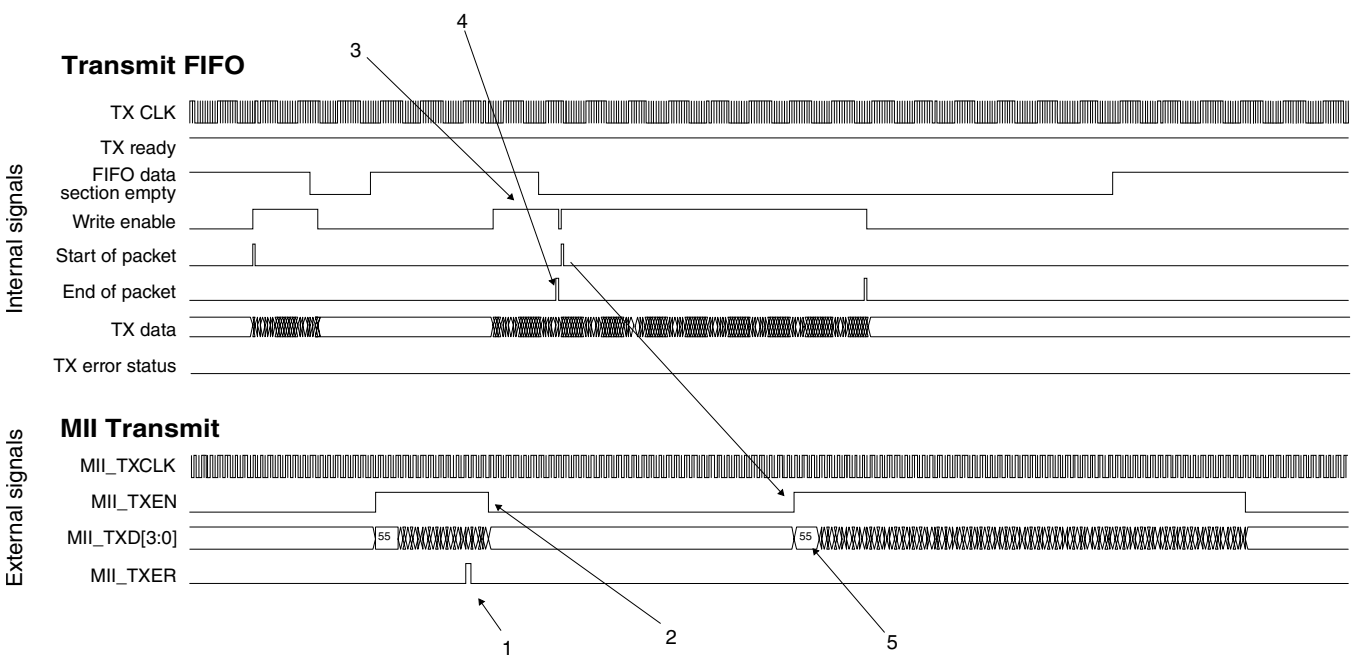


Figure 41-16. Transmit FIFO underflow protection

41.3.16.2 Transmit FIFO overflow

On the transmit path, when the FIFO reaches the programmable almost full threshold, the internal MAC ready signal is deasserted. The application should stop sending new data.

However, if the application keeps sending data, the transmit FIFO overflows, corrupting contents that were previously stored. The core logic sets the enhanced TxBD[OE] field for the next frame transmitted to indicate this overflow occurence.

Note

Overflow is a fatal error and must be addressed by resetting the core or clearing ENET*n*_ECR[ETHER_EN], to clear the FIFOs and prepare for normal operation again.

41.3.16.3 Receive FIFO overflow

During a frame reception, if the client application is not able to receive data (1), the MAC receive control truncates the incoming frame when the FIFO reaches the programmable almost-full threshold to avoid an overflow.

The frame is subsequently received on the FIFO interface with an error indication (enhanced RxBD[ME] field set together with receive end-of-packet) (2) with the truncation error status field set (3).

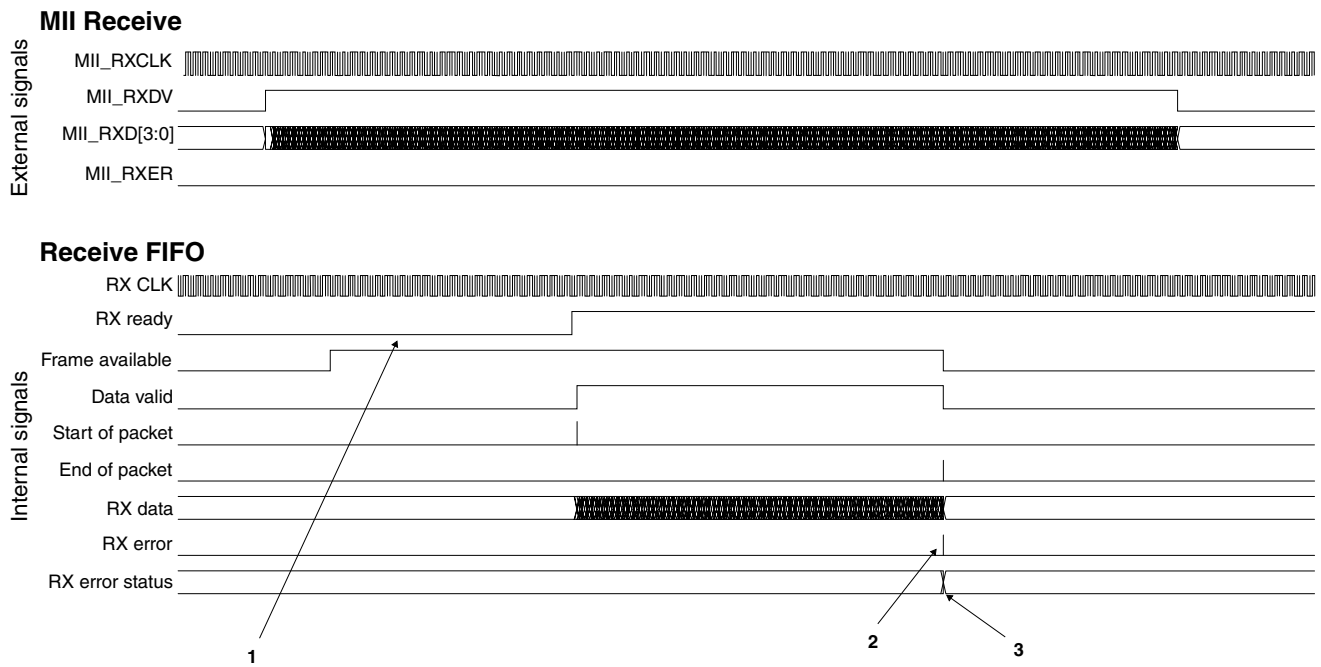


Figure 41-17. Receive FIFO overflow protection

41.3.17 PHY management interface

The MDIO interface is a two-wire management interface. The MDIO management interface implements a standardized method to access the PHY device management registers.

The core implements a master MDIO interface, which can be connected to up to 32 PHY devices.

41.3.17.1 MDIO clause 22 frame format

The core MDIO master controller communicates with the slave (PHY device) using frames that are defined in the following table.

A complete frame has a length of 64 bits made up of an optional 32-bit preamble, 14-bit command, 2-bit bus direction change, and 16-bit data. Each bit is transferred on the rising edge of the MDIO clock (MDC signal). The MDIO data signal is tri-stated between frames.

The core PHY management interface supports the standard MDIO specification (IEEE 802.3 Clause 22).

Table 41-38. MDIO clause 22 frame structure

ST	OP	PHYADR	REGADR	TA	DATA

Table 41-39. MDIO frame field descriptions

Field	Description
ST (2 bits)	Start indication field, programmed with ENET*n*_MMFR[ST] and equal to 01 for Standard MDIO (Clause 22).
OP (2 bits)	Opcode defines type of operation. Programmed with ENET*n*_MMFR[OP]. 01 Write operation 10 Read operation
PHYADR (5 bits)	Five-bit PHY device address, programmed with ENET*n*_MMFR[PA]. Up to 32 devices can be addressed.
REGADR (5 bits)	Five-bit register address, programmed with ENET*n*_MMFR[RA]. Each PHY can implement up to 32 registers.
TA (2 bits)	Turnaround time, programmed with ENET*n*_MMFR[TA]. Two bit-times are reserved for read operations to switch the data bus from write to read. The PHY device presents its register contents in the data phase and drives the bus from the second bit of the turnaround phase.
Data	Data, set by ENET*n*_MMFR[DATA]. Written to or read from the PHY

Table 41-39. MDIO frame field descriptions

Field	Description
(16 bits)	

41.3.17.2 MDIO clause 45 frame format

The extended MDIO frame structure defined in IEEE 802.3 Clause 45 introduces indirect addressing. First, a write transaction to an address register is done, followed by a write or read transaction which will put the 16-bit data in the register or retrieve the register contents respectively. A preamble of 32 bits of logical ones is sent prior to every transaction. The MDIO data signal is tri-stated between frames.

The extended MDIO defines four transactions, which are determined by the two-bit opcode field.

Table 41-40. MDIO clause 45 frame structure

ST	OP	PRTAD	DEVAD	TA	ADDR/DATA

All bits are transmitted from left to right (Preamble bits first) and all fields have their Most-Significant bit sent first (leftmost in above table). The 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).

The fields and transactions are summarized in the following tables.

Table 41-41. MDIO clause 45 frame field descriptions

Field	Description
ST	Start indication. Indicates the end of the preamble and start of the frame. This value is 00 for extended MDIO (Clause 45) frames.
OP	Opcode defines if a read or write operation is performed and is programmed with ENET*n*_MMFR[OP]. See Table 41-42 for more information. 00 Address write 01 Write operation 10 Read inc. operation 11 Read operation
PRTAD	The port address specifies a MDIO port. Each Port can have up to 32 devices which each can have a separate set of registers.
DEVAD	Device address. Up to 32 devices can be addressed (within a port).
TA	Turnaround time, programmed with ENET*n*_MMFR[TA]. Two bit-times are reserved for read operations to switch the data bus from write to read. The PHY device presents its register contents in the data phase and drives the bus from the second bit of the turnaround phase.

Table continues on the next page...

Table 41-41. MDIO clause 45 frame field descriptions (continued)

Field	Description
ADDR/DATA	16-bit address (for address write) or data, set by ENET*n*_MMFR[DATA], written to or read from the PHY.

Table 41-42. MDIO Clause 45 Transactions

Transaction Type	Description
Address	A write transaction to the internal address register of the device/port. The data section of the frame contains the value to be stored in the device's internal address "pointer" register for further transactions.
Write	Data write to a register. The 16 bit data will be written to the register identified by the device-internal address.
Read	Data is read from the register identified by the device-internal address.
Read inc.	Read with address postincrement. The register identified by the device-internal address is read. After this, the device-internal address is incremented. If the address register is all '1' (0xFFFF) no increment is done (i.e. increment does not wrap around).

41.3.17.3 MDIO clock generation

The MDC clock is generated from the internal bus clock (i.e., IPS bus clock) divided by the value programmed in ENET*n*_MSCR[MII_SPEED].

41.3.17.4 MDIO operation

To perform an MDIO access, set the MDIO command register (ENET*n*_MMFR) according to the description provided in MII Management Frame Register (ENETn_MMFR).

To check when the programmed access completes, read the ENET*n*_EIR[MII] field.

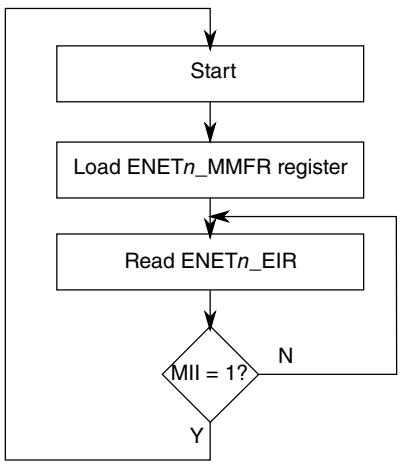


Figure 41-18. MDIO access overview

41.3.18 Ethernet interfaces

The following Ethernet interfaces are implemented:

- Fast Ethernet MII 10/100 (Media Independent Interface)
- RMII 10/100 using interface converters/gaskets

The following table shows how to configure ENET registers to select each interface.

Mode	RCR[RMII_10T]	RCR[RMII_MODE]
MII - 10 Mbit/s	—	0
MII - 100 Mbit/s	—	0
RMII - 10 Mbit/s	1	1
RMII - 100 Mbit/s	0	1

41.3.18.1 RMII interface

In RMII receive mode, for normal reception following assertion of CRS_DV, RXD[1:0] is 00 until the receiver determines that the receive event has a proper start-of-stream delimiter (SSD).

The preamble appears (RXD[1:0]=01) and the MACs begin capturing data following detection of SFD.

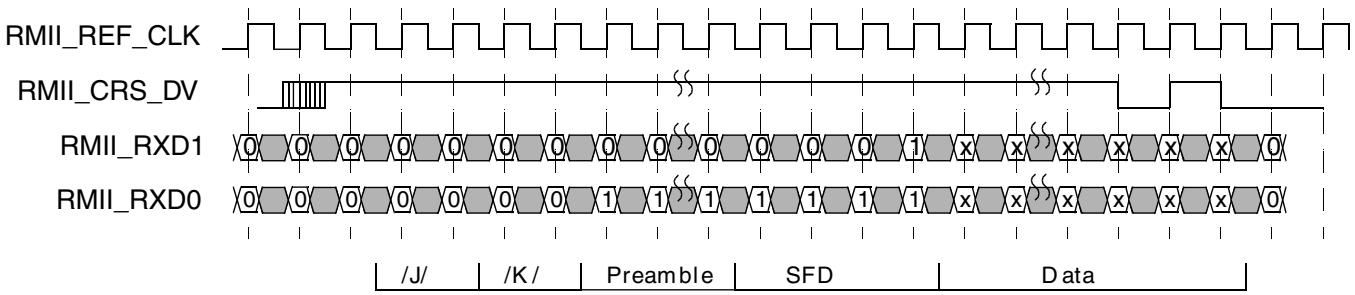


Figure 41-19. RMII receive operation

If a false carrier is detected (bad SSD), then RXD[1:0] is 10 until the end of the receive event. This is a unique pattern since a false carrier can only occur at the beginning of a packet where the preamble is decoded (RXD[1:0] = 01).

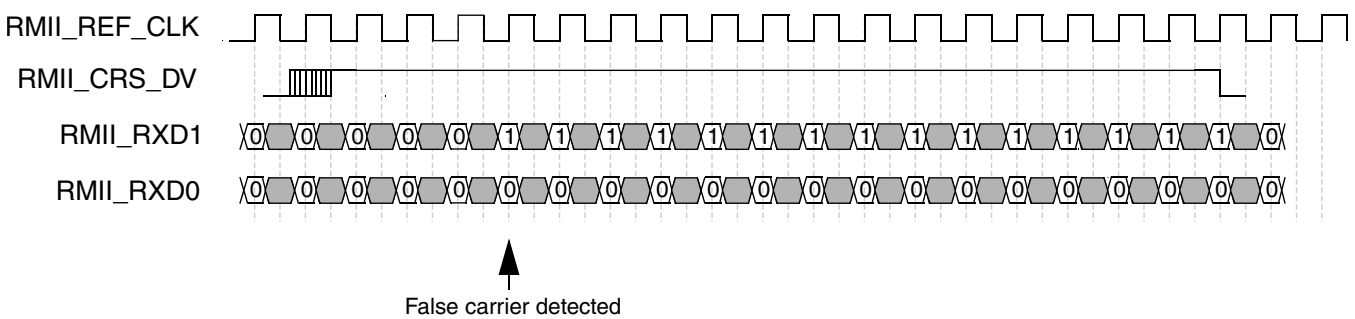


Figure 41-20. RMII receive operation with false carrier

In RMII transmit mode, TXD[1:0] provides valid data for each REF_CLK period while TXEN is asserted.

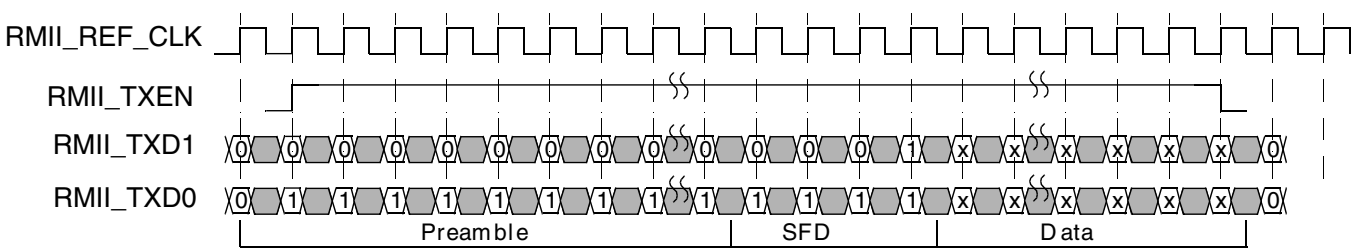


Figure 41-21. RMII transmit operation

41.3.18.2 MII Interface — transmit

On transmit, all data transfers are synchronous to MII_TXCLK rising edge. The MII data enable signal MII_TXEN is asserted to indicate the start of a new frame, and remains asserted until the last byte of the frame is present on the MII_TXD[3:0] bus.

Between frames, MII_TXEN remains deasserted.

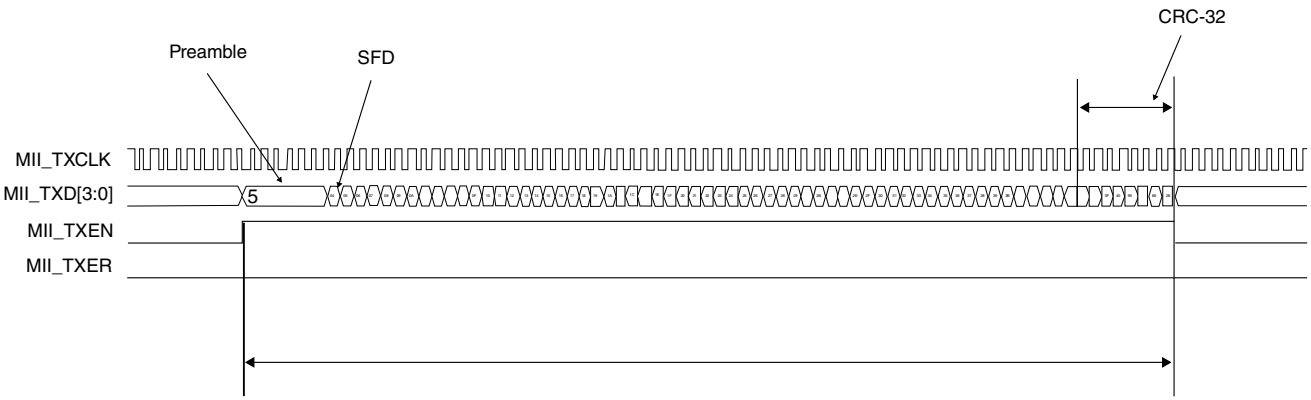


Figure 41-22. MII transmit operation

If a frame is received on the FIFO interface with an error (for example, RxBD[ME] set) the frame is subsequently transmitted with the MII_TXER error signal for one clock cycle at any time during the packet transfer.

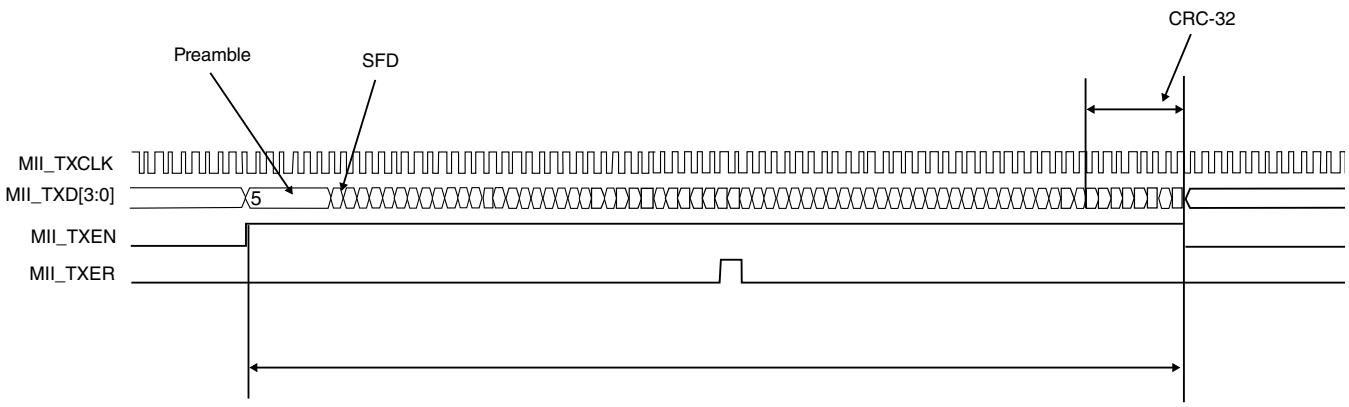


Figure 41-23. MII transmit operation — errored frame

41.3.18.2.1 Transmit with collision — half-duplex

When a collision is detected during a frame transmission (MII_COL asserted), the MAC stops the current transmission, sends a 32-bit jam pattern, and re-transmits the current frame.

(See Collision detection in half-duplex mode for details)

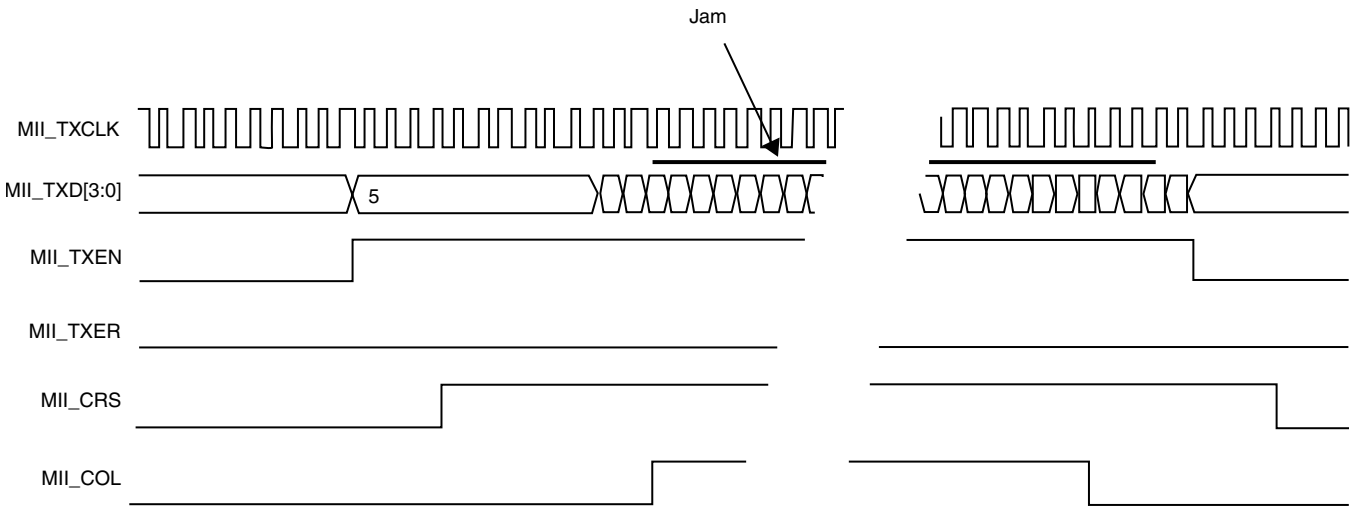


Figure 41-24. MII transmit operation — transmission with collision

41.3.18.3 MII interface — receive

On receive, all signals are sampled on the MII_RXCLK rising edge. The MII data enable signal, MII_RXDV, is asserted by the PHY to indicate the start of a new frame and remains asserted until the last byte of the frame is present on MII_RXD[3:0] bus.

Between frames, MII_RXDV remains deasserted.

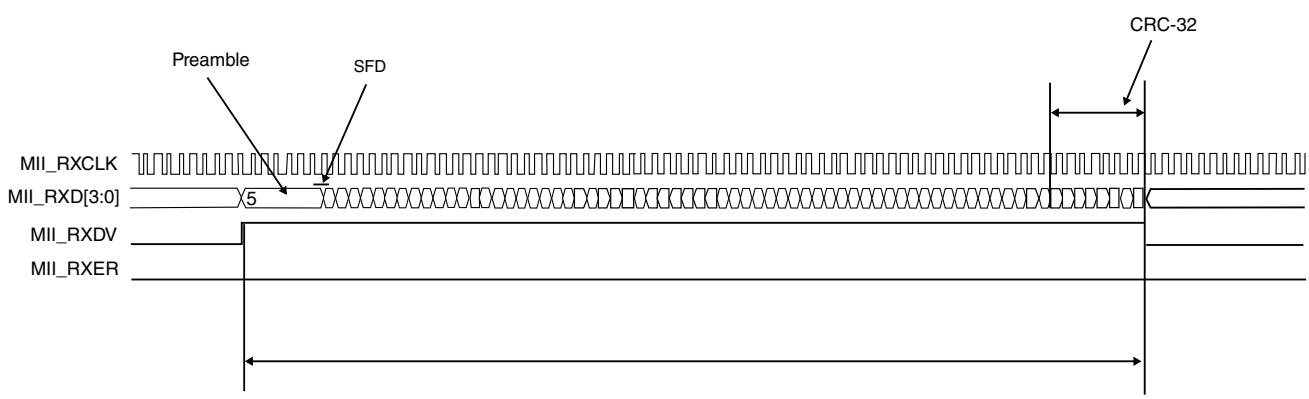


Figure 41-25. MII receive operation

If the PHY detects an error on the frame received from the line, the PHY asserts the MII error signal, MII_RXER, for at least one clock cycle at any time during the packet transfer.

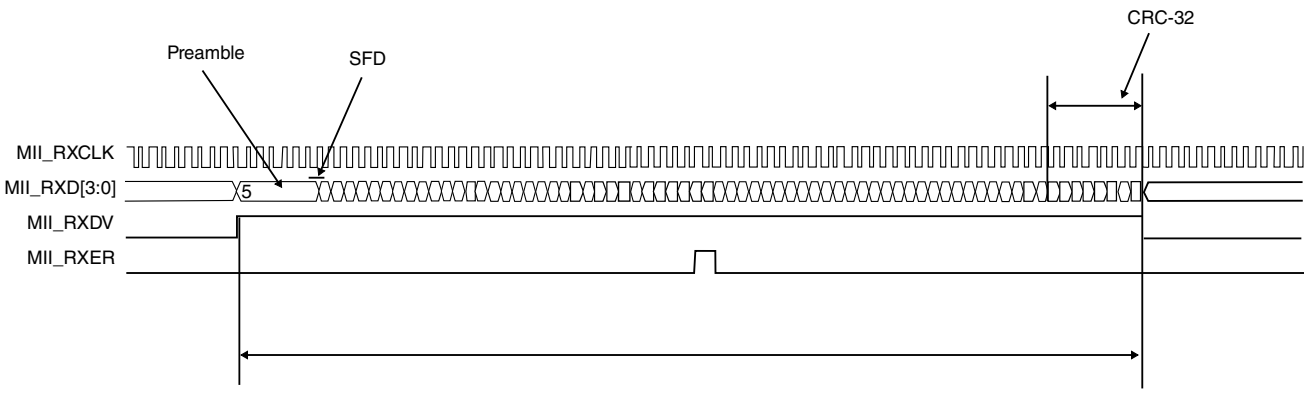


Figure 41-26. MII receive operation — errored frame

A frame received on the MII interface with a PHY error indication is subsequently transferred on the FIFO interface with RxBD[ME] set.

41.3.19 Interrupt coalescence

The purpose of the interrupt coalescing is to reduce the number of interrupts generated by the MAC so as to reduce the CPU loading.

To facilitate this interrupt coalescing, these registers are available with the same control and configuration fields.

- Transmit Interrupt Coalescing Register (TXICn)
- Receive Interrupt Coalescing Register (RXICn)

When coalescing is enabled by asserting the corresponding ICEN field and such interrupt is also enabled by the corresponding interrupt mask of the EIMR register, the MAC generates an interrupt when the threshold number of frames is reached (defined by ICFT) or when the threshold timer expires (defined by ICTT).

When coalescing is disabled by de-asserting ICEN, but interrupt is enabled by the corresponding interrupt mask of the EIMR register, the MAC generates an interrupt as they are received without using coalescing. Interrupt coalescing is done for each transmit and receive queue/class independently.

41.3.19.1 Interrupt coalescence setup

Interrupt coalescence supports both legacy and enhanced BDs. The following guidelines are recommended when setting up interrupt coalescence.

- When the MAC is configured for enhanced (IEEE 1588) mode, that is, enhanced BDs:
  - Set the INT bit in the enhanced received buffer descriptor to one.
  - Set the INT bit in the enhanced transmit buffer descriptor(s) to one.
- Clear the TXB and RXB fields in the EIMR register.

41.3.19.2 Updating the frame count threshold on-the-fly

To update the ICFT field in the RXIC and TXIC registers:

1. Disable interrupt coalescence by clearing the appropriate ICEN field. This will allow the internal interrupt coalescence counter to reset to zero.

NOTE

When disabling interrupt coalescence, if an interrupt event is pending, that is, the interrupt counter is not zero, then an interrupt will occur.

2. Write the new threshold value to the ICFT field.
3. Set ICEN to one.

NOTE

The ICFT field can be updated on-the-fly without disabling the ICEN field. The hardware interrupt will continue and there is a possibility that an interrupt will occur depending on the state of the hardware counter and the previous ICFT value.

41.3.19.3 Updating the timer threshold on-the-fly

To update the ICTT field in the RXIC and TXIC registers:

1. Disable interrupt coalescence by clearing the appropriate ICEN field. This will allow the internal interrupt coalescence counter to reset to zero.

NOTE

When disabling interrupt coalescence, if an interrupt event is pending, that is, the interrupt counter is not zero, then an interrupt will occur.

2. Write the new timer value to the ICTT field.
3. Set ICEN to one.

41.3.20 Clocks

The table found here describes the clock sources for ENET. Please see the chip-specific clocking section for clock setting, configuration and gating information.

Table 41-43. Clocks

Clock name	Description
ipg_clk	Module clock
ipg_clk_rmii	Module clock for RMII
ipg_clk_s	Peripheral access clock
ipg_clk_time (ts_clk)	Peripheral clock
txc_sampling_clk	Transmit sampling clock

41.4 External Signals

The table found here describes the external signals of ENET.

Table 41-44. ENET External Signals

Signal	Description	Mode	Direction
1588_EVENT_IN	Capture/compare block input/ output event bus signal. When configured for capture and a rising edge is detected, the current timer value is latched and transferred into the corresponding ENET_TCCRn register for inspection by software. When configured for compare, the corresponding signal 1588_EVENT is asserted for one cycle when the timer reaches the compare value programmed in register ENET_TCCRn. An interrupt or DMA request can be triggered if the corresponding bit in ENET_TCSRn[TIE] or ENET_TCSRn[TDRE] is set.	MII/RMII	I
1588_EVENT_OUT	Capture/compare block input/ output event bus signal. When configured for capture and a rising edge is detected, the current timer value is latched and transferred into the	MII/RMII	O

Table continues on the next page...

Table 41-44. ENET External Signals (continued)

Signal	Description	Mode	Direction
	corresponding ENET_TCCRn register for inspection by software. When configured for compare, the corresponding signal 1588_EVENT is asserted for one cycle when the timer reaches the compare value programmed in register ENET_TCCRn. An interrupt or DMA request can be triggered if the corresponding bit in ENET_TCSRn[TIE] or ENET_TCSRn[TDRE] is set. NOTE: ENET_1588_EVENT0_OUT has a programmable output width, see IOMUXC_GPR[CLK_STRETCH], delayed one clock cycle in relation to all other EVENTx_OUT signals.		
COL	Asserted upon detection of a collision and remains asserted while the collision persists. This signal is not defined for full duplex mode.	MII	I
CRS	Carrier sense. When asserted, indicates transmit or receive medium is not idle. In RMII mode, this signal is present on the CRS_DV pin.	MII	I
MDC	Output clock provides a timing reference to the PHY for data transfers on the MDIO signal.	MII/RMII	O
MDIO	Transfers control information between the external PHY and the media access controller. Data is synchronous to MDC. This signal is an input after reset.	MII/RMII	IO
RX_CLK	In MII mode, provides a timing reference for RX_EN, RX_DATA[3:0], and RX_ER.	MII	I
RX_ER	When asserted with RXDV, indicates the PHY detects an error in the current frame.	MII/RMII	I
RX_EN	Asserting this input indicates the PHY has valid nibbles present on the MII. RX_EN must remain asserted from the first recovered nibble of the frame through to the last	MII/RMII	I

Table continues on the next page...

Table 41-44. ENET External Signals (continued)

Signal	Description	Mode	Direction
	nibble. Asserting RX_EN must start no later than the SFD and exclude any EOF. In RMII mode, this pin also generates the CRS signal.		
TX_CLK	Input clock, which provides a timing reference for TX_EN, TX_DATA[3:0], and TX_ER.	MII	O
TX_ER	When asserted for one or more clock cycles while TXEN is also asserted, PHY sends one or more illegal symbols.	MII/RMII	O
TX_EN	Indicates when valid nibbles are present on the MII. This signal is asserted with the first nibble of a preamble and is deasserted before the first TX_CLK following the final nibble of the frame.	MII/RMII	I
RDATA	Contains the Ethernet input data transferred from the PHY to the media-access controller when RX_EN is asserted.	MII/RMII	I
TDATA	Serial output Ethernet data. Only valid during TX_EN assertion.	MII/RMII	O
REF_CLK1	In RMII mode, this signal is the reference clock for receive, transmit, and the control interface.	RMII	IO

41.5 Memory map/register definition

ENET registers must be read or written with 32-bit accesses. Non-32 bit accesses will terminate with an error.

Reserved bits should be written with 0 and ignored on read. Unused registers read zero and a write has no effect.

This table shows Ethernet registers organization.

Table 41-45. Register map summary

Offset Address	Section	Description
0x0000 – 0x01FF	Configuration	Core control and status registers

Table continues on the next page...

Table 41-45. Register map summary (continued)

Offset Address	Section	Description
0x0200 – 0x03FF	Statistics counters	MIB and Remote Network Monitoring (RFC 2819) registers
0x0400 – 0x0430	1588 control	1588 adjustable timer (TSM) and 1588 frame control
0x0600 – 0x07FC	Capture/Compare block	Registers for the Capture/Compare block

41.5.1 ENET register descriptions

41.5.1.1 ENET memory map

ENET base address: 402D_8000h

ENET2 base address: 402D_4000h

Offset	Register	Width (In bits)	Access	Reset value
4h	Interrupt Event Register (EIR)	32	WOZ	0000_0000h
8h	Interrupt Mask Register (EIMR)	32	RW	0000_0000h
10h	Receive Descriptor Active Register - Ring 0 (RDAR)	32	RW	0000_0000h
14h	Transmit Descriptor Active Register - Ring 0 (TDAR)	32	RW	0000_0000h
24h	Ethernet Control Register (ECR)	32	RW	7000_0000h
40h	MII Management Frame Register (MMFR)	32	RW	0000_0000h
44h	MII Speed Control Register (MSCR)	32	RW	0000_0000h
64h	MIB Control Register (MIBC)	32	RW	C000_0000h
84h	Receive Control Register (RCR)	32	RW	05EE_0001h
C4h	Transmit Control Register (TCR)	32	RW	0000_0000h
E4h	Physical Address Lower Register (PALR)	32	RW	0000_0000h
E8h	Physical Address Upper Register (PAUR)	32	RW	0000_8808h
ECh	Opcode/Pause Duration Register (OPD)	32	RW	0001_0000h
F0h	Transmit Interrupt Coalescing Register (TXIC0)	32	RW	0000_0000h
100h	Receive Interrupt Coalescing Register (RXIC0)	32	RW	0000_0000h
118h	Descriptor Individual Upper Address Register (IAUR)	32	RW	0000_0000h
11Ch	Descriptor Individual Lower Address Register (IALR)	32	RW	0000_0000h
120h	Descriptor Group Upper Address Register (GAUR)	32	RW	0000_0000h
124h	Descriptor Group Lower Address Register (GALR)	32	RW	0000_0000h
144h	Transmit FIFO Watermark Register (TFWR)	32	RW	0000_0000h
180h	Receive Descriptor Ring 0 Start Register (RDSR)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
184h	Transmit Buffer Descriptor Ring 0 Start Register (TDSR)	32	RW	0000_0000h
188h	Maximum Receive Buffer Size Register - Ring 0 (MRBR)	32	RW	0000_0000h
190h	Receive FIFO Section Full Threshold (RSFL)	32	RW	0000_0000h
194h	Receive FIFO Section Empty Threshold (RSEM)	32	RW	0000_0000h
198h	Receive FIFO Almost Empty Threshold (RAEM)	32	RW	0000_0004h
19Ch	Receive FIFO Almost Full Threshold (RAFL)	32	RW	0000_0004h
1A0h	Transmit FIFO Section Empty Threshold (TSEM)	32	RW	0000_0000h
1A4h	Transmit FIFO Almost Empty Threshold (TAEM)	32	RW	0000_0004h
1A8h	Transmit FIFO Almost Full Threshold (TAFL)	32	RW	0000_0008h
1ACh	Transmit Inter-Packet Gap (TIPG)	32	RW	0000_000Ch
1B0h	Frame Truncation Length (FTRL)	32	RW	0000_07FFh
1C0h	Transmit Accelerator Function Configuration (TACC)	32	RW	0000_0000h
1C4h	Receive Accelerator Function Configuration (RACC)	32	RW	0000_0000h
204h	Tx Packet Count Statistic Register (RMON_T_PACKETS)	32	RO	0000_0000h
208h	Tx Broadcast Packets Statistic Register (RMON_T_BC_PKT)	32	RO	0000_0000h
20Ch	Tx Multicast Packets Statistic Register (RMON_T_MC_PKT)	32	RO	0000_0000h
210h	Tx Packets with CRC/Align Error Statistic Register (RMON_T_CRC_ALIGN)	32	RO	0000_0000h
214h	Tx Packets Less Than Bytes and Good CRC Statistic Register (RMON_T_UNDERSIZE)	32	RO	0000_0000h
218h	Tx Packets GT MAX_FL bytes and Good CRC Statistic Register (RMON_T_OVERSIZE)	32	RO	0000_0000h
21Ch	Tx Packets Less Than 64 Bytes and Bad CRC Statistic Register (RMON_T_FRAG)	32	RO	0000_0000h
220h	Tx Packets Greater Than MAX_FL bytes and Bad CRC Statistic Register (RMON_T_JAB)	32	RO	0000_0000h
224h	Tx Collision Count Statistic Register (RMON_T_COL)	32	RO	0000_0000h
228h	Tx 64-Byte Packets Statistic Register (RMON_T_P64)	32	RO	0000_0000h
22Ch	Tx 65- to 127-byte Packets Statistic Register (RMON_T_P65TO127)	32	RO	0000_0000h
230h	Tx 128- to 255-byte Packets Statistic Register (RMON_T_P128TO255)	32	RO	0000_0000h
234h	Tx 256- to 511-byte Packets Statistic Register (RMON_T_P256TO511)	32	RO	0000_0000h
238h	Tx 512- to 1023-byte Packets Statistic Register (RMON_T_P512TO1023)	32	RO	0000_0000h
23Ch	Tx 1024- to 2047-byte Packets Statistic Register (RMON_T_P1024TO2047)	32	RO	0000_0000h
240h	Tx Packets Greater Than 2048 Bytes Statistic Register (RMON_T_P_GTE2048)	32	RO	0000_0000h
244h	Tx Octets Statistic Register (RMON_T_OCTETS)	32	RO	0000_0000h
24Ch	Frames Transmitted OK Statistic Register (IEEE_T_FRAME_OK)	32	RO	0000_0000h
250h	Frames Transmitted with Single Collision Statistic Register (IEEE_T_1COL)	32	RO	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
254h	Frames Transmitted with Multiple Collisions Statistic Register (IEEE_T_MCOL)	32	RO	0000_0000h
258h	Frames Transmitted after Deferral Delay Statistic Register (IEEE_T_DEF)	32	RO	0000_0000h
25Ch	Frames Transmitted with Late Collision Statistic Register (IEEE_T_LCOL)	32	RO	0000_0000h
260h	Frames Transmitted with Excessive Collisions Statistic Register (IEEE_T_EXCOL)	32	RO	0000_0000h
264h	Frames Transmitted with Tx FIFO Underrun Statistic Register (IEEE_T_MACERR)	32	RO	0000_0000h
268h	Frames Transmitted with Carrier Sense Error Statistic Register (IEEE_T_CSERR)	32	RO	0000_0000h
26Ch	Reserved Statistic Register (IEEE_T_SQE)	32	RO	0000_0000h
270h	Flow Control Pause Frames Transmitted Statistic Register (IEEE_T_FDXFC)	32	RO	0000_0000h
274h	Octet Count for Frames Transmitted w/o Error Statistic Register (IEEE_T_OCTETS_OK)	32	RO	0000_0000h
284h	Rx Packet Count Statistic Register (RMON_R_PACKETS)	32	RO	0000_0000h
288h	Rx Broadcast Packets Statistic Register (RMON_R_BC_PKT)	32	RO	0000_0000h
28Ch	Rx Multicast Packets Statistic Register (RMON_R_MC_PKT)	32	RO	0000_0000h
290h	Rx Packets with CRC/Align Error Statistic Register (RMON_R_CRC_ALIGN)	32	RO	0000_0000h
294h	Rx Packets with Less Than 64 Bytes and Good CRC Statistic Register (RMON_R_UNDERSIZE)	32	RO	0000_0000h
298h	Rx Packets Greater Than MAX_FL and Good CRC Statistic Register (RMON_R_OVERSIZE)	32	RO	0000_0000h
29Ch	Rx Packets Less Than 64 Bytes and Bad CRC Statistic Register (RMON_R_FRAG)	32	RO	0000_0000h
2A0h	Rx Packets Greater Than MAX_FL Bytes and Bad CRC Statistic Register (RMON_R_JAB)	32	RO	0000_0000h
2A8h	Rx 64-Byte Packets Statistic Register (RMON_R_P64)	32	RO	0000_0000h
2ACh	Rx 65- to 127-Byte Packets Statistic Register (RMON_R_P65TO127)	32	RO	0000_0000h
2B0h	Rx 128- to 255-Byte Packets Statistic Register (RMON_R_P128TO255)	32	RO	0000_0000h
2B4h	Rx 256- to 511-Byte Packets Statistic Register (RMON_R_P256TO511)	32	RO	0000_0000h
2B8h	Rx 512- to 1023-Byte Packets Statistic Register (RMON_R_P512TO1023)	32	RO	0000_0000h
2BCh	Rx 1024- to 2047-Byte Packets Statistic Register (RMON_R_P1024TO2047)	32	RO	0000_0000h
2C0h	Rx Packets Greater than 2048 Bytes Statistic Register (RMON_R_P_GTE2048)	32	RO	0000_0000h
2C4h	Rx Octets Statistic Register (RMON_R_OCTETS)	32	RO	0000_0000h
2C8h	Frames not Counted Correctly Statistic Register (IEEE_R_DROP)	32	RO	0000_0000h
2CCh	Frames Received OK Statistic Register (IEEE_R_FRAME_OK)	32	RO	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
2D0h	Frames Received with CRC Error Statistic Register (IEEE_R_CRC)	32	RO	0000_0000h
2D4h	Frames Received with Alignment Error Statistic Register (IEEE_R_ALIGN)	32	RO	0000_0000h
2D8h	Receive FIFO Overflow Count Statistic Register (IEEE_R_MACERR)	32	RO	0000_0000h
2DCh	Flow Control Pause Frames Received Statistic Register (IEEE_R_FDXFC)	32	RO	0000_0000h
2E0h	Octet Count for Frames Received without Error Statistic Register (IEEE_R_OCTETS_OK)	32	RO	0000_0000h
400h	Adjustable Timer Control Register (ATCR)	32	RW	0000_0000h
404h	Timer Value Register (ATVR)	32	RW	0000_0000h
408h	Timer Offset Register (ATOFF)	32	RW	0000_0000h
40Ch	Timer Period Register (ATPER)	32	RW	3B9A_CA00h
410h	Timer Correction Register (ATCOR)	32	RW	0000_0000h
414h	Time-Stamping Clock Period Register (ATINC)	32	RW	0000_0000h
418h	Timestamp of Last Transmitted Frame (ATSTMP)	32	RO	0000_0000h
604h	Timer Global Status Register (TGSR)	32	W1C	0000_0000h
608h	Timer Control Status Register (TCSR0)	32	RW	0000_0000h
60Ch	Timer Compare Capture Register (TCCR0)	32	RW	0000_0000h
610h	Timer Control Status Register (TCSR1)	32	RW	0000_0000h
614h	Timer Compare Capture Register (TCCR1)	32	RW	0000_0000h
618h	Timer Control Status Register (TCSR2)	32	RW	0000_0000h
61Ch	Timer Compare Capture Register (TCCR2)	32	RW	0000_0000h
620h	Timer Control Status Register (TCSR3)	32	RW	0000_0000h
624h	Timer Compare Capture Register (TCCR3)	32	RW	0000_0000h

41.5.1.2 Interrupt Event Register (EIR)

41.5.1.2.1 Offset

Register	Offset
EIR	4h

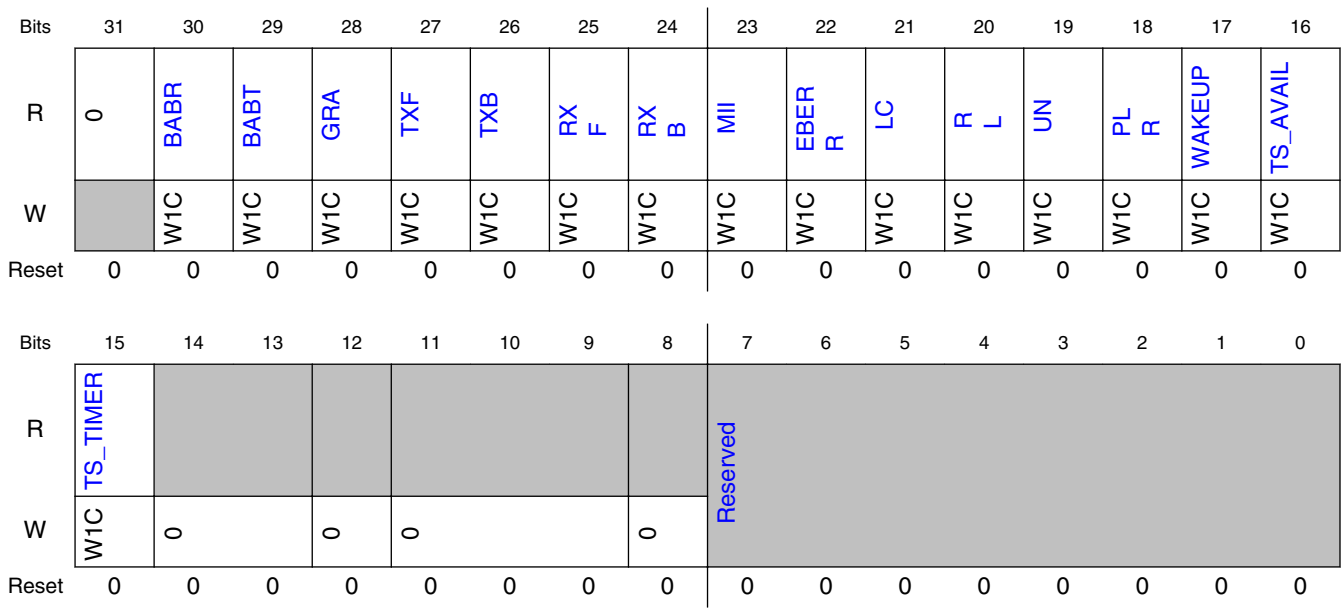
41.5.1.2.2 Function

When an event occurs that sets a bit in EIR, an interrupt occurs if the corresponding bit in the interrupt mask register (EIMR) is also set. Writing a 1 to an EIR bit clears it; writing 0 has no effect. This register is cleared upon hardware reset.

NOTE

TxBD[INT] and RxBD[INT] must be set to 1 to allow setting the corresponding EIR register flags in enhanced mode, ENET_ECR[EN1588] = 1. Legacy mode does not require these flags to be enabled.

41.5.1.2.3 Diagram



41.5.1.2.4 Fields

Field	Function
31 —	Reserved
30 BABR	Babbling Receive Error Indicates a frame was received with length in excess of RCR[MAX_FL] bytes.
29 BABT	Babbling Transmit Error Indicates the transmitted frame length exceeds RCR[MAX_FL] bytes. Usually this condition is caused when a frame that is too long is placed into the transmit data buffer(s). Truncation does not occur.
28 GRA	Graceful Stop Complete This interrupt is asserted after the transmitter is put into a pause state after completion of the frame currently being transmitted. See Graceful Transmit Stop (GTS) for conditions that lead to graceful stop. **NOTE:** The GRA interrupt is asserted only when the TX transitions into the stopped state. If this bit is cleared by writing 1 and the TX is still stopped, the bit is not set again.
27 TXF	Transmit Frame Interrupt Indicates a frame has been transmitted and the last corresponding buffer descriptor has been updated.

Table continues on the next page...

Field	Function
26 TXB	Transmit Buffer Interrupt Indicates a transmit buffer descriptor has been updated.
25 RXF	Receive Frame Interrupt Indicates a frame has been received and the last corresponding buffer descriptor has been updated.
24 RXB	Receive Buffer Interrupt Indicates a receive buffer descriptor is not the last in the frame has been updated.
23 MII	MII Interrupt. Indicates that the MII has completed the data transfer requested.
22 EBERR	Ethernet Bus Error Indicates a system bus error occurred when a uDMA transaction is underway. When this bit is set, ECR[ETHEREN] is cleared, halting frame processing by the MAC. When this occurs, software must ensure proper actions, possibly resetting the system, to resume normal operation.
21 LC	Late Collision Indicates a collision occurred beyond the collision window (slot time) in half-duplex mode. The frame truncates with a bad CRC and the remainder of the frame is discarded.
20 RL	Collision Retry Limit Indicates a collision occurred on each of 16 successive attempts to transmit the frame. The frame is discarded without being transmitted and transmission of the next frame commences. This error can only occur in half-duplex mode.
19 UN	Transmit FIFO Underrun Indicates the transmit FIFO became empty before the complete frame was transmitted. **NOTE:** In situations where the device has various masters generating high traffic, a FIFO underrun can occur on the transmit FIFO. To avoid transmit FIFO underrun, store and forward can be enabled in ENET_TFWR[STRFWD]. See Transmit FIFO Watermark Register (TFWR). Also, a higher priority can be set for ENET traffic using available means on the central bus fabric connecting the ENET module.
18 PLR	Payload Receive Error Indicates a frame was received with a payload length error. See Frame Length/Type Verification: Payload Length Check for more information.
17 WAKEUP	Node Wakeup Request Indication Read-only status bit to indicate that a magic packet has been detected. Will act only if ECR[MAGICEN] is set.
16 TS_AVAIL	Transmit Timestamp Available Indicates that the timestamp of the last transmitted timing frame is available in the ATSTMP register.
15 TS_TIMER	Timestamp Timer The adjustable timer reached the period event. A period event interrupt can be generated if ATCR[PEREN] is set and the timer wraps according to the periodic setting in the ATPER register. Set the timer period value before setting ATCR[PEREN].
14-13 —	This write-only field is reserved. It must always be written with the value 0.
12 —	This write-only field is reserved. It must always be written with the value 0.
11-9	This write-only field is reserved. It must always be written with the value 0.

Table continues on the next page...

Field	Function
—	
8 —	This write-only field is reserved. It must always be written with the value 0.
7-0 —	This write-only field is reserved. It must always be written with the value 0.

41.5.1.3 Interrupt Mask Register (EIMR)

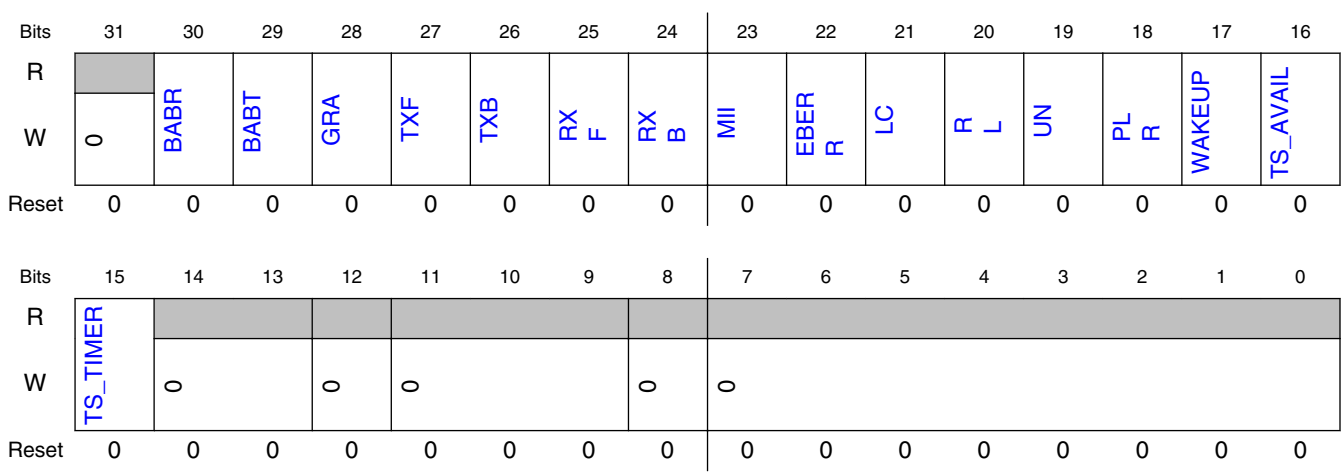
41.5.1.3.1 Offset

Register	Offset
EIMR	8h

41.5.1.3.2 Function

EIMR controls which interrupt events are allowed to generate actual interrupts. A hardware reset clears this register. If the corresponding bits in the EIR and EIMR registers are set, an interrupt is generated. The interrupt signal remains asserted until a 1 is written to the EIR field (write 1 to clear) or a 0 is written to the EIMR field.

41.5.1.3.3 Diagram



41.5.1.3.4 Fields

Field	Function
31 —	This write-only field is reserved. It must always be written with the value 0.
30 BABR	BABR Interrupt Mask Corresponds to interrupt source EIR[BABR] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR BABR field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared. 0b - The corresponding interrupt source is masked. 1b - The corresponding interrupt source is not masked.
29 BABT	BABT Interrupt Mask Corresponds to interrupt source EIR[BABT] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR BABT field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared. 0b - The corresponding interrupt source is masked. 1b - The corresponding interrupt source is not masked.
28 GRA	GRA Interrupt Mask Corresponds to interrupt source EIR[GRA] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR GRA field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared. 0b - The corresponding interrupt source is masked. 1b - The corresponding interrupt source is not masked.
27 TXF	TXF Interrupt Mask Corresponds to interrupt source EIR[TXF] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR TXF field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared. 0b - The corresponding interrupt source is masked. 1b - The corresponding interrupt source is not masked.
26 TXB	TXB Interrupt Mask Corresponds to interrupt source EIR[TXB] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR TXF field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared. 0b - The corresponding interrupt source is masked. 1b - The corresponding interrupt source is not masked.
25 RXF	RXF Interrupt Mask Corresponds to interrupt source EIR[RXF] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR RXF field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared. 0b - The corresponding interrupt source is masked. 1b - The corresponding interrupt source is not masked.
24 RXB	RXB Interrupt Mask Corresponds to interrupt source EIR[RXB] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The

Table continues on the next page...

Field	Function
	corresponding EIR RXB field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared. 0b - The corresponding interrupt source is masked. 1b - The corresponding interrupt source is not masked.
23 MII	MII Interrupt Mask Corresponds to interrupt source EIR[MII] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR MII field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared. 0b - The corresponding interrupt source is masked. 1b - The corresponding interrupt source is not masked.
22 EBERR	EBERR Interrupt Mask Corresponds to interrupt source EIR[EBERR] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR EBERR field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared. 0b - The corresponding interrupt source is masked. 1b - The corresponding interrupt source is not masked.
21 LC	LC Interrupt Mask Corresponds to interrupt source EIR[LC] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR LC field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared. 0b - The corresponding interrupt source is masked. 1b - The corresponding interrupt source is not masked.
20 RL	RL Interrupt Mask Corresponds to interrupt source EIR[RL] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR RL field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared. 0b - The corresponding interrupt source is masked. 1b - The corresponding interrupt source is not masked.
19 UN	UN Interrupt Mask Corresponds to interrupt source EIR[UN] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR UN field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared. 0b - The corresponding interrupt source is masked. 1b - The corresponding interrupt source is not masked.
18 PLR	PLR Interrupt Mask Corresponds to interrupt source EIR[PLR] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR PLR field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared. 0b - The corresponding interrupt source is masked. 1b - The corresponding interrupt source is not masked.
17 WAKEUP	WAKEUP Interrupt Mask Corresponds to interrupt source EIR[WAKEUP] register and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR WAKEUP field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared.

Table continues on the next page...

Field	Function
	0b - The corresponding interrupt source is masked. 1b - The corresponding interrupt source is not masked.
16 TS_AVAIL	TS_AVAIL Interrupt Mask Corresponds to interrupt source EIR[TS_AVAIL] register and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR TS_AVAIL field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared. 0b - The corresponding interrupt source is masked. 1b - The corresponding interrupt source is not masked.
15 TS_TIMER	TS_TIMER Interrupt Mask Corresponds to interrupt source EIR[TS_TIMER] register and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR TS_TIMER field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared. 0b - The corresponding interrupt source is masked. 1b - The corresponding interrupt source is not masked.
14-13 —	This write-only field is reserved. It must always be written with the value 0.
12 —	This write-only field is reserved. It must always be written with the value 0.
11-9 —	This write-only field is reserved. It must always be written with the value 0.
8 —	This write-only field is reserved. It must always be written with the value 0.
7-0 —	This write-only field is reserved. It must always be written with the value 0.

41.5.1.4 Receive Descriptor Active Register - Ring 0 (RDAR)

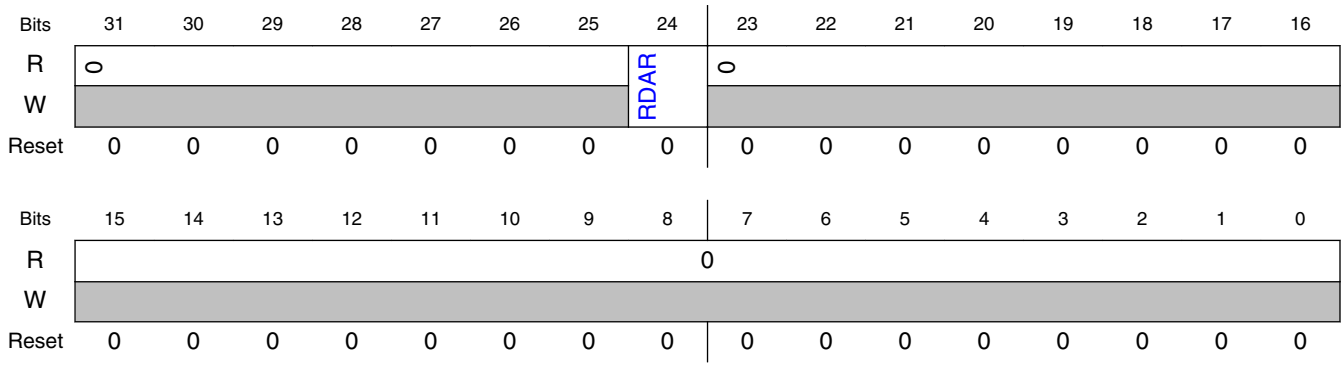
41.5.1.4.1 Offset

Register	Offset
RDAR	10h

41.5.1.4.2 Function

RDAR is a command register, written by the user, to indicate that the receive descriptor ring has been updated, that is, that the driver produced empty receive buffers with the empty bit set.

41.5.1.4.3 Diagram



41.5.1.4.4 Fields

Field	Function
31-25 —	Reserved
24 RDAR	Receive Descriptor Active Always set to 1 when this register is written, regardless of the value written. This field is cleared by the MAC device when no additional empty descriptors remain in the receive ring. It is also cleared when ECR[ETHEREN] transitions from set to cleared or when ECR[RESET] is set.
23-0 —	Reserved

41.5.1.5 Transmit Descriptor Active Register - Ring 0 (TDAR)

41.5.1.5.1 Offset

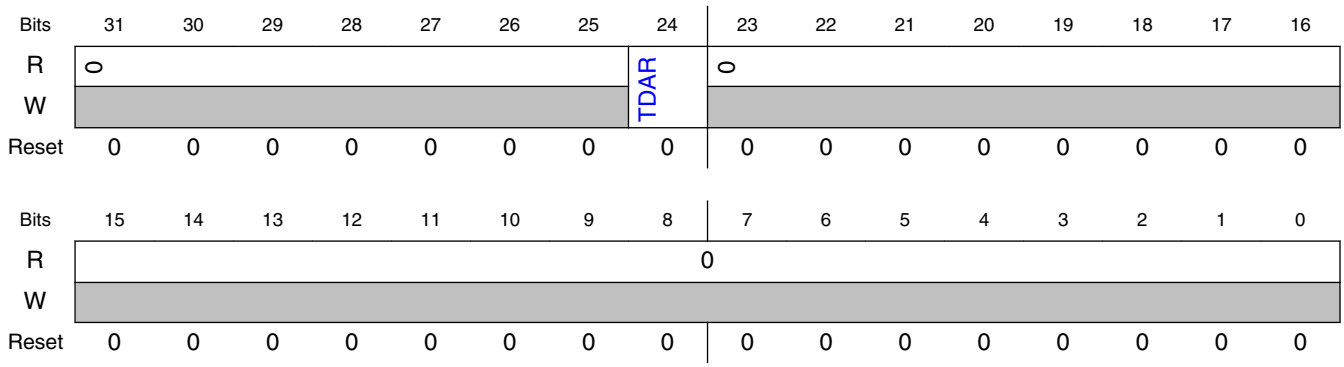
Register	Offset
TDAR	14h

41.5.1.5.2 Function

The TDAR is a command register that the user writes to indicate that the transmit descriptor ring has been updated, that is, that transmit buffers have been produced by the driver with the ready bit set in the buffer descriptor.

The TDAR register is cleared at reset, when ECR[ETHEREN] transitions from set to cleared, or when ECR[RESET] is set.

41.5.1.5.3 Diagram



41.5.1.5.4 Fields

Field	Function
31-25 —	Reserved
24 TDAR	Transmit Descriptor Active Always set to 1 when this register is written, regardless of the value written. This bit is cleared by the MAC device when no additional ready descriptors remain in the transmit ring. Also cleared when ECR[ETHEREN] transitions from set to cleared or when ECR[RESET] is set.
23-0 —	Reserved

41.5.1.6 Ethernet Control Register (ECR)

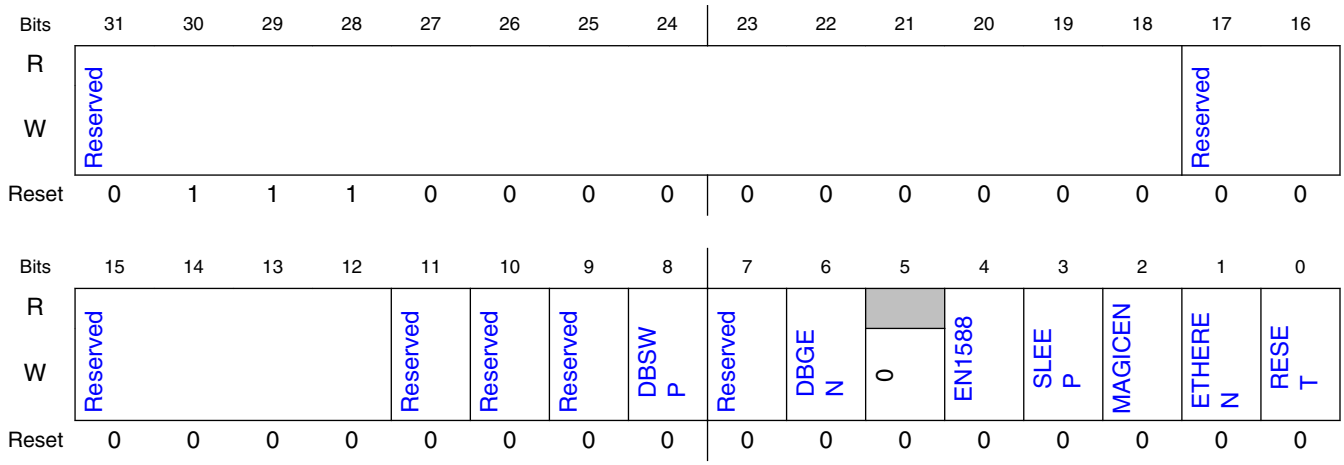
41.5.1.6.1 Offset

Register	Offset
ECR	24h

41.5.1.6.2 Function

ECR is a read/write user register, though hardware may also alter fields in this register. It controls many of the high level features of the Ethernet MAC, including legacy FEC support through the EN1588 field.

41.5.1.6.3 Diagram



41.5.1.6.4 Fields

Field	Function
31-18 —	Always write 01110000000000b to this field.
17-12 —	Always write 0 to this field.
11 —	Always write 0 to this field.
10 —	Always write 0 to this field.
9 —	Always write 0 to this field.
8 DBSWP	Descriptor Byte Swapping Enable Swaps the byte locations of the buffer descriptors. **NOTE:** This field must be written to 1 after reset. 0b - The buffer descriptor bytes are not swapped to support big-endian devices. 1b - The buffer descriptor bytes are swapped to support little-endian devices.
7 —	Always write 0 to this field.
6 DBGEN	Debug Enable Enables the MAC to enter hardware freeze mode when the device enters debug mode. 0b - MAC continues operation in debug mode. 1b - MAC enters hardware freeze mode when the processor is in debug mode.
5 —	This write-only field is reserved. It must always be written with the value 0.
4	EN1588 Enable

Table continues on the next page...

Field	Function
EN1588	Enables enhanced functionality of the MAC. 0b - Legacy FEC buffer descriptors and functions enabled. 1b - Enhanced frame time-stamping functions enabled. Has no effect within the MAC besides controlling the DMA control bit ena_1588.
3 SLEEP	Sleep Mode Enable 0b - Normal operating mode. 1b - Sleep mode.
2 MAGICEN	Magic Packet Detection Enable Enables/disables magic packet detection. **NOTE:** MAGICEN is relevant only if the SLEEP field is set. If MAGICEN is set, changing the SLEEP field enables/disables sleep mode and magic packet detection. **NOTE:** EIMR[WAKEUP] must be written to one if Magic packet wakeup is programed to wake up the chip from low power mode. 0b - Magic detection logic disabled. 1b - The MAC core detects magic packets and asserts EIR[WAKEUP] when a frame is detected.
1 ETHEREN	Ethernet Enable Enables/disables the Ethernet MAC. When the MAC is disabled, the buffer descriptors for an aborted transmit frame are not updated. The uDMA, buffer descriptor, and FIFO control logic are reset, including the buffer descriptor and FIFO pointers. Hardware clears this field under the following conditions: • RESET is set by software • An error condition causes the EBERR field to set. **NOTE:** • ETHEREN must be set at the very last step during ENET configuration/setup/initialization, only *after* all other ENET-related registers have been configured. • If ETHEREN is cleared to 0 by software then next time ETHEREN is set, the EIR interrupts must cleared to 0 due to previous pending interrupts. 0b - Reception immediately stops and transmission stops after a bad CRC is appended to any currently transmitted frame. 1b - MAC is enabled, and reception and transmission are possible.
0 RESET	Ethernet MAC Reset When this field is set, it clears the ETHEREN field.

41.5.1.7 MII Management Frame Register (MMFR)

41.5.1.7.1 Offset

Register	Offset
MMFR	40h

41.5.1.7.2 Function

Writing to MMFR triggers a management frame transaction to the PHY device unless MSCR is programmed to zero.

If MSCR is changed from zero to non-zero during a write to MMFR, an MII frame is generated with the data previously written to the MMFR. This allows MMFR and MSCR to be programmed in either order if MSCR is currently zero.

If the MMFR register is written while frame generation is in progress, the frame contents are altered. Software must use the EIR[MII] interrupt indication to avoid writing to the MMFR register while frame generation is in progress.

41.5.1.7.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ST		OP		PA					RA					TA	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

41.5.1.7.4 Fields

Field	Function
31-30 ST	Start Of Frame Delimiter See Table 41-38 (Clause 22) or Table 41-40 (Clause 45) for correct value.
29-28 OP	Operation Code See Table 41-38 (Clause 22) or Table 41-40 (Clause 45) for correct value.
27-23 PA	PHY Address See Table 41-38 (Clause 22) or Table 41-40 (Clause 45) for correct value.
22-18 RA	Register Address See Table 41-38 (Clause 22) or Table 41-40 (Clause 45) for correct value.
17-16 TA	Turn Around This field must be programmed to 10 to generate a valid MII management frame.
15-0 DATA	Management Frame Data This is the field for data to be written to or read from the PHY register.

41.5.1.8 MII Speed Control Register (MSCR)

41.5.1.8.1 Offset

Register	Offset
MSCR	44h

41.5.1.8.2 Function

MSCR provides control of the MII clock (MDC pin) frequency and allows a preamble drop on the MII management frame.

The MII_SPEED field must be programmed with a value to provide an MDC frequency of less than or equal to 2.5 MHz to be compliant with the IEEE 802.3 MII specification. The MII_SPEED must be set to a non-zero value to source a read or write management frame. After the management frame is complete, the MSCR register may optionally be cleared to turn off MDC. The MDC signal generated has a 50% duty cycle except when MII_SPEED changes during operation. This change takes effect following a rising or falling edge of MDC.

For example, if the internal module clock (that is, peripheral bus clock) is 25 MHz, programming MII_SPEED to 0x4 results in an MDC as given in the following equation:

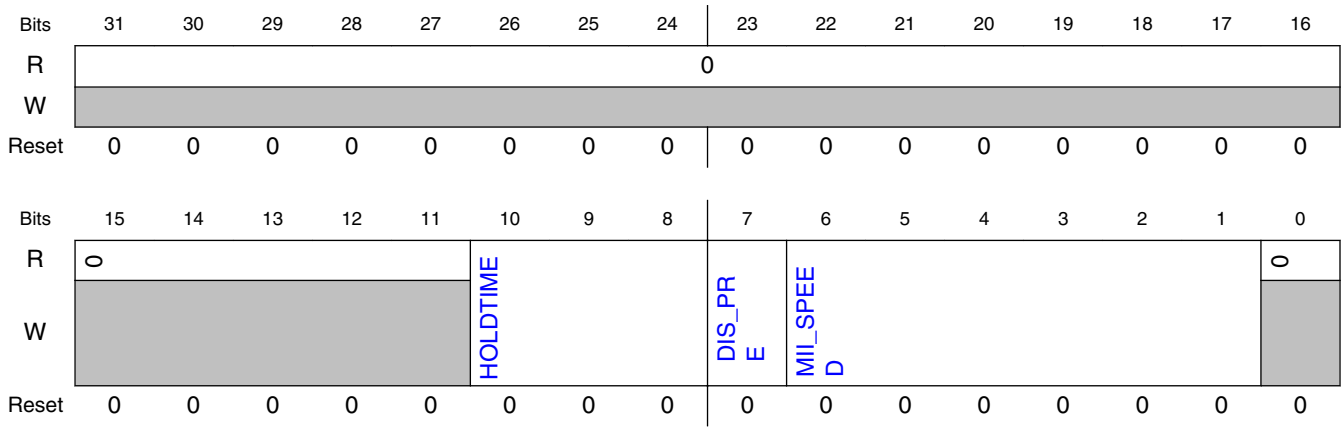
MII clock frequency = 25 MHz / ((4 + 1) x 2) = 2.5 MHz

The following table shows the optimum values for MII_SPEED as a function of IPS bus clock frequency.

Table 41-46. Programming Examples for MSCR

Internal module clock frequency	MSCR [MII_SPEED]	MDC frequency
25 MHz	0x4	2.50 MHz
33 MHz	0x6	2.36 MHz
40 MHz	0x7	2.50 MHz
50 MHz	0x9	2.50 MHz
66 MHz	0xD	2.36 MHz

41.5.1.8.3 Diagram



41.5.1.8.4 Fields

Field	Function
31-11 —	Reserved
10-8 HOLDTIME	Hold time On MDIO Output IEEE802.3 clause 22 defines a minimum of 10 ns for the hold time on the MDIO output. Depending on the host bus frequency, the setting may need to be increased. 000b - 1 internal module clock cycle 001b - 2 internal module clock cycles 010b - 3 internal module clock cycles 111b - 8 internal module clock cycles
7 DIS_PRE	Disable Preamble Enables/disables prepending a preamble to the MII management frame. The MII standard allows the preamble to be dropped if the attached PHY devices do not require it. 0b - Preamble enabled. 1b - Preamble (32 ones) is not prepended to the MII management frame.
6-1 MII_SPEED	MII Speed Controls the frequency of the MII management interface clock (MDC) relative to the internal module clock. A value of 0 in this field turns off MDC and leaves it in low voltage state. Any non-zero value results in the MDC frequency of: 1/((MII_SPEED + 1) x 2) of the internal module clock frequency
0 —	Reserved

41.5.1.9 MIB Control Register (MIBC)

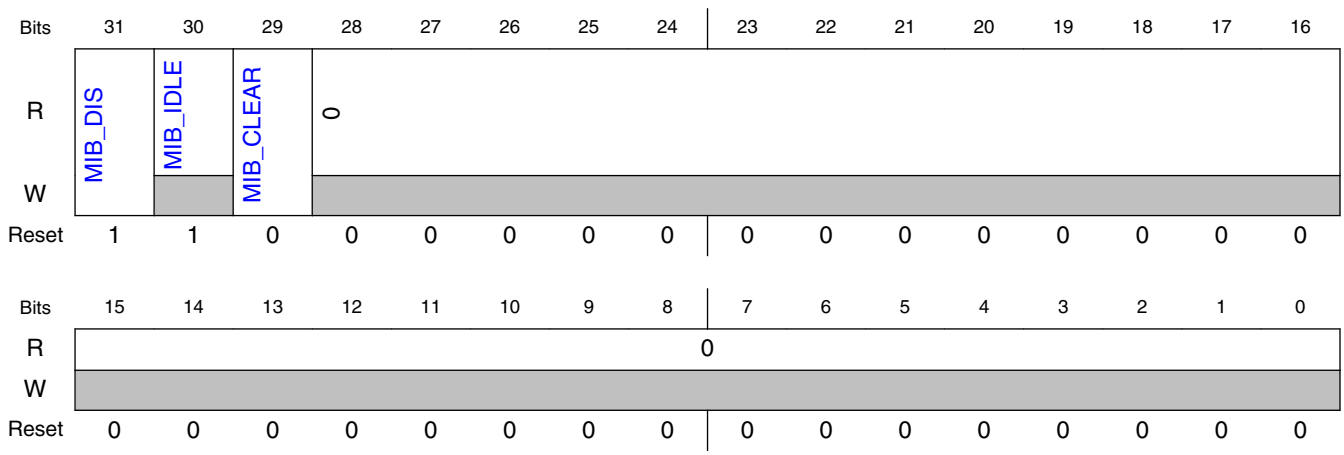
41.5.1.9.1 Offset

Register	Offset
MIBC	64h

41.5.1.9.2 Function

MIBC is a read/write register controlling and observing the state of the MIB block. Access this register to disable the MIB block operation or clear the MIB counters. The MIB_DIS field resets to 1.

41.5.1.9.3 Diagram



41.5.1.9.4 Fields

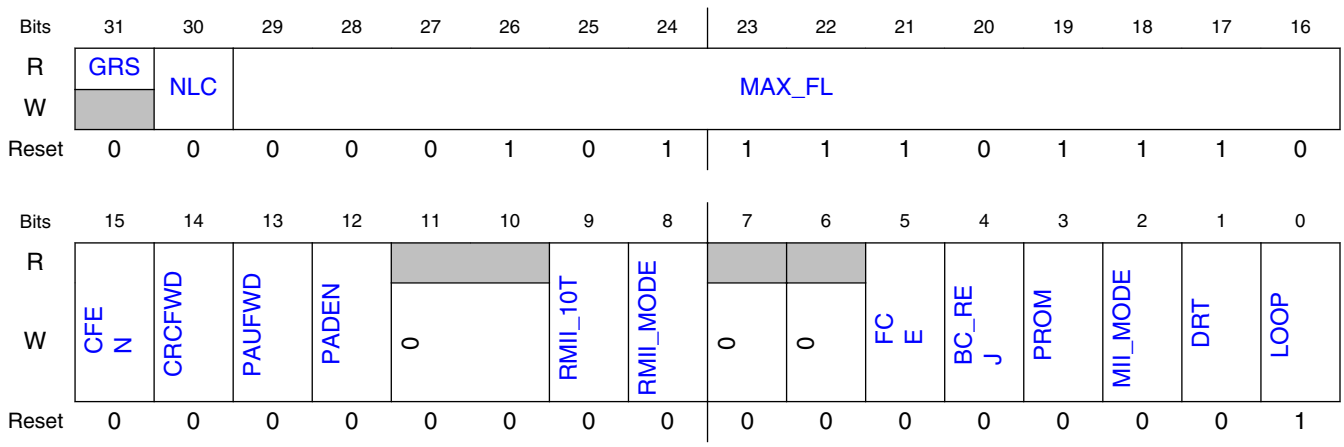
Field	Function
31 MIB_DIS	Disable MIB Logic If this control field is set, 0b - MIB logic is enabled. 1b - MIB logic is disabled. The MIB logic halts and does not update any MIB counters.
30 MIB_IDLE	MIB Idle 0b - The MIB block is updating MIB counters. 1b - The MIB block is not currently updating any MIB counters.
29 MIB_CLEAR	MIB Clear **NOTE:** This field is not self-clearing. To clear the MIB counters set and then clear this field. 0b - See note above. 1b - All statistics counters are reset to 0.
28-0 —	Reserved

41.5.1.10 Receive Control Register (RCR)

41.5.1.10.1 Offset

Register	Offset
RCR	84h

41.5.1.10.2 Diagram



41.5.1.10.3 Fields

Field	Function
31 GRS	Graceful Receive Stopped Read-only status indicating that the MAC receive datapath is stopped. 0b - Receive not stopped 1b - Receive stopped
30 NLC	Payload Length Check Disable Enables/disables a payload length check. 0b - The payload length check is disabled. 1b - The core checks the frame's payload length with the frame length/type field. Errors are indicated in the EIR[PLR] field.
29-16 MAX_FL	Maximum Frame Length Resets to decimal 1518. Length is measured starting at DA and includes the CRC at the end of the frame. Transmit frames longer than MAX_FL cause the BABT interrupt to occur. Receive frames longer than MAX_FL cause the BABR interrupt to occur and set the LG field in the end of frame receive buffer descriptor. The recommended default value to be programmed is 1518 or 1522 if VLAN tags are supported.
15	MAC Control Frame Enable

Table continues on the next page...

Field	Function
CFEN	Enables/disables the MAC control frame. 0b - MAC control frames with any opcode other than 0x0001 (pause frame) are accepted and forwarded to the client interface. 1b - MAC control frames with any opcode other than 0x0001 (pause frame) are silently discarded.
14 CRCFWD	Terminate/Forward Received CRC Specifies whether the CRC field of received frames is transmitted or stripped. **NOTE:** If padding function is enabled (PADEN = 1), CRCFWD is ignored and the CRC field is checked and always terminated and removed. 0b - The CRC field of received frames is transmitted to the user application. 1b - The CRC field is stripped from the frame.
13 PAUFWD	Terminate/Forward Pause Frames Specifies whether pause frames are terminated or forwarded. 0b - Pause frames are terminated and discarded in the MAC. 1b - Pause frames are forwarded to the user application.
12 PADEN	Enable Frame Padding Remove On Receive Specifies whether the MAC removes padding from received frames. 0b - No padding is removed on receive by the MAC. 1b - Padding is removed from received frames.
11-10 —	This write-only field is reserved. It must always be written with the value 0.
9 RMII_10T	Enables 10-Mbit/s mode of the RMII . 0b - 100-Mbit/s operation. 1b - 10-Mbit/s operation.
8 RMII_MODE	RMII Mode Enable Specifies whether the MAC is configured for MII mode or RMII operation . 0b - MAC configured for MII mode. 1b - MAC configured for RMII operation.
7 —	This write-only field is reserved. It must always be written with the value 0.
6 —	This write-only field is reserved. It must always be written with the value 0.
5 FCE	Flow Control Enable If set, the receiver detects PAUSE frames. Upon PAUSE frame detection, the transmitter stops transmitting data frames for a given duration. 0b - Disable flow control 1b - Enable flow control
4 BC_REJ	Broadcast Frame Reject If set, frames with destination address (DA) equal to 0xFFFF_FFFF_FFFF are rejected unless the PROM field is set. If BC_REJ and PROM are set, frames with broadcast DA are accepted and the MISS (M) is set in the receive buffer descriptor. 0b - Will not reject frames as described above 1b - Will reject frames as described above
3 PROM	Promiscuous Mode All frames are accepted regardless of address matching. 0b - Disabled. 1b - Enabled.

Table continues on the next page...

Field	Function
2 MII_MODE	Media Independent Interface Mode This field must always be set. 0b - Reserved. 1b - MII or RMII mode, as indicated by the RMII_MODE field.
1 DRT	Disable Receive On Transmit 0b - Receive path operates independently of transmit (i.e., full-duplex mode). Can also be used to monitor transmit activity in half-duplex mode. 1b - Disable reception of frames while transmitting. (Normally used for half-duplex mode.)
0 LOOP	Internal Loopback This is an MII internal loopback, therefore MII_MODE must be written to 1 and RMII_MODE must be written to 0. 0b - Loopback disabled. 1b - Transmitted frames are looped back internal to the device and transmit MII output signals are not asserted. DRT must be cleared.

41.5.1.11 Transmit Control Register (TCR)

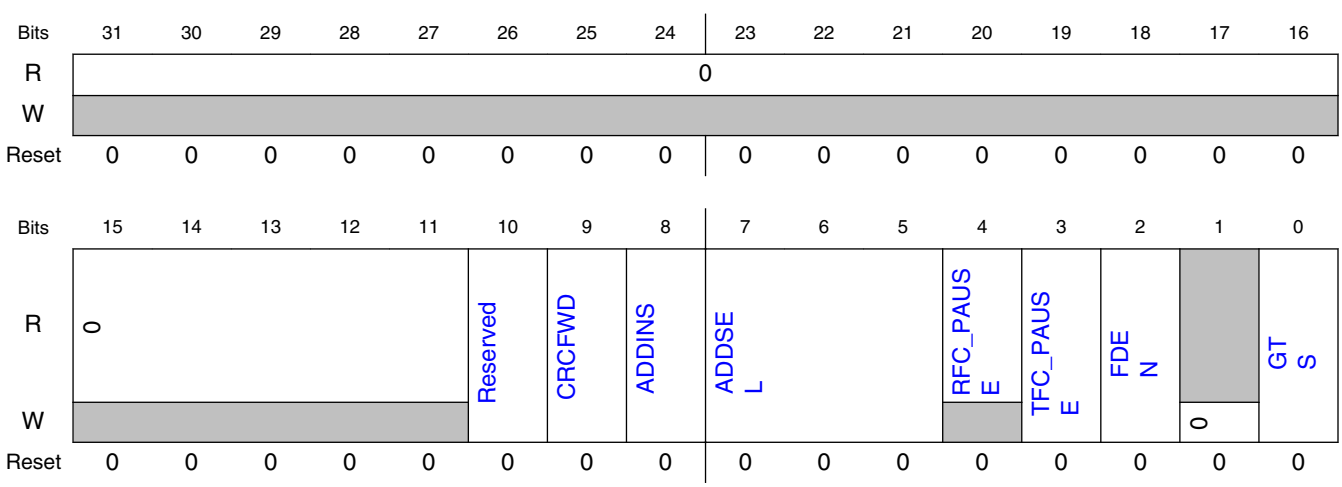
41.5.1.11.1 Offset

Register	Offset
TCR	C4h

41.5.1.11.2 Function

TCR is read/write and configures the transmit block. This register is cleared at system reset. FDEN can only be modified when ECR[ETHEREN] is cleared.

41.5.1.11.3 Diagram



41.5.1.11.4 Fields

Field	Function
31-11 —	Reserved
10 —	This field is read/write and must be set to 0.
9 CRCFWD	Forward Frame From Application With CRC 0b - TxBD[TC] controls whether the frame has a CRC from the application. 1b - The transmitter does not append any CRC to transmitted frames, as it is expecting a frame with CRC from the application.
8 ADDINS	Set MAC Address On Transmit 0b - The source MAC address is not modified by the MAC. 1b - The MAC overwrites the source MAC address with the programmed MAC address according to ADDSEL.
7-5 ADDSEL	Source MAC Address Select On Transmit If ADDINS is set, indicates the MAC address that overwrites the source MAC address. 000b - Node MAC address programmed on PADDR1/2 registers. 100b - Reserved. 101b - Reserved. 110b - Reserved.
4 RFC_PAUSE	Receive Frame Control Pause This status field is set when a full-duplex flow control pause frame is received and the transmitter pauses for the duration defined in this pause frame. This field automatically clears when the pause duration is complete.
3 TFC_PAUSE	Transmit Frame Control Pause Pauses frame transmission. When this field is set, EIR[GRA] is set. With transmission of data frames stopped, the MAC transmits a MAC control PAUSE frame. Next, the MAC clears TFC_PAUSE and resumes transmitting data frames. If the transmitter pauses due to user assertion of GTS or reception of a PAUSE frame, the MAC may continue transmitting a MAC control PAUSE frame. 0b - No PAUSE frame transmitted. 1b - The MAC stops transmission of data frames after the current transmission is complete.
2 FDEN	Full-Duplex Enable If this field is set, frames transmit independent of carrier sense and collision inputs. Only modify this bit when ECR[ETHEREN] is cleared. 0b - Disable full-duplex 1b - Enable full-duplex
1 —	This write-only field is reserved. It must always be written with the value 0.
0 GTS	Graceful Transmit Stop When this field is set, MAC stops transmission after any frame currently transmitted is complete and EIR[GRA] is set. If frame transmission is not currently underway, the GRA interrupt is asserted immediately. After transmission finishes, clear GTS to restart. The next frame in the transmit FIFO is then transmitted. If an early collision occurs during transmission when GTS is set, transmission stops after the collision. The frame is transmitted again after GTS is cleared. There may be old frames in the transmit FIFO that transmit when GTS is reasserted. To avoid this, clear ECR[ETHEREN] following the GRA interrupt.

Field	Function
	0b - Disable graceful transmit stop 1b - Enable graceful transmit stop

41.5.1.12 Physical Address Lower Register (PALR)

41.5.1.12.1 Offset

Register	Offset
PALR	E4h

41.5.1.12.2 Function

PALR contains the lower 32 bits (bytes 0, 1, 2, 3) of the 48-bit address used in the address recognition process to compare with the destination address (DA) field of receive frames with an individual DA. In addition, this register is used in bytes 0 through 3 of the six-byte source address field when transmitting PAUSE frames.

41.5.1.12.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PADDR1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PADDR1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

41.5.1.12.4 Fields

Field	Function
31-0 PADDR1	Pause Address Bytes 0 (bits 31:24), 1 (bits 23:16), 2 (bits 15:8), and 3 (bits 7:0) of the 6-byte individual address are used for exact match and the source address field in PAUSE frames.

41.5.1.13 Physical Address Upper Register (PAUR)

41.5.1.13.1 Offset

Register	Offset
PAUR	E8h

41.5.1.13.2 Function

PAUR contains the upper 16 bits (bytes 4 and 5) of the 48-bit address used in the address recognition process to compare with the destination address (DA) field of receive frames with an individual DA. In addition, this register is used in bytes 4 and 5 of the six-byte source address field when transmitting PAUSE frames. Bits 15:0 of PAUR contain a constant type field (0x8808) for transmission of PAUSE frames.

41.5.1.13.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PADDR2															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TYPE															
W																
Reset	1	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0

41.5.1.13.4 Fields

Field	Function
31-16 PADDR2	Bytes 4 (bits 31:24) and 5 (bits 23:16) of the 6-byte individual address used for exact match, and the source address field in PAUSE frames.
15-0 TYPE	Type Field In PAUSE Frames These fields have a constant value of 0x8808.

41.5.1.14 Opcode/Pause Duration Register (OPD)

41.5.1.14.1 Offset

Register	Offset
OPD	ECh

41.5.1.14.2 Function

OPD is read/write accessible. This register contains the 16-bit opcode and 16-bit pause duration fields used in transmission of a PAUSE frame. The opcode field is a constant value, 0x0001. When another node detects a PAUSE frame, that node pauses transmission for the duration specified in the pause duration field. The lower 16 bits of this register are not reset and you must initialize it.

41.5.1.14.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	OPCODE															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PAUSE_DUR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

41.5.1.14.4 Fields

Field	Function
31-16 OPCODE	Opcode Field In PAUSE Frames These fields have a constant value of 0x0001.
15-0 PAUSE_DUR	Pause Duration Pause duration field used in PAUSE frames.

41.5.1.15 Transmit Interrupt Coalescing Register (TXIC0)

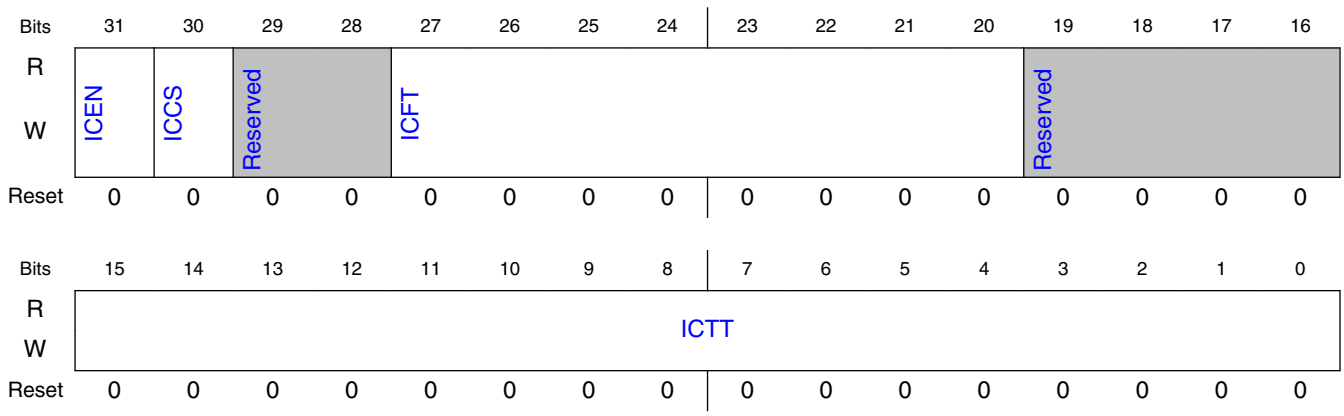
41.5.1.15.1 Offset

Register	Offset
TXIC0	F0h

41.5.1.15.2 Function

See Interrupt coalescence for more information.

41.5.1.15.3 Diagram



41.5.1.15.4 Fields

Field	Function
31 ICEN	Interrupt Coalescing Enable 0b - Disable Interrupt coalescing. 1b - Enable Interrupt coalescing.
30 ICCS	Interrupt Coalescing Timer Clock Source Select 0b - Use MII/GMII TX clocks. 1b - Use ENET system clock.
29-28 —	This field must be set to 0.
27-20 ICFT	Interrupt coalescing frame count threshold This value determines the number of frames needed to be transmitted for raising an interrupt. Frame counter restarts after reaching this threshold value or after the expiring of the coalescing timer. Must be greater than zero to avoid unpredictable behavior.
19-16 —	This field must be set to 0.
15-0 ICTT	Interrupt coalescing timer threshold

Field	Function
	Interrupt coalescing timer threshold in units of 64 clock periods. This value determines the maximum amount of time after transmitting a frame before raising an interrupt. The threshold timer is disabled after expiring or number of frame transmission defined by ICFT and starts again upon transmission of the next first frame. Must be greater than zero to avoid unpredictable behavior.

41.5.1.16 Receive Interrupt Coalescing Register (RXIC0)

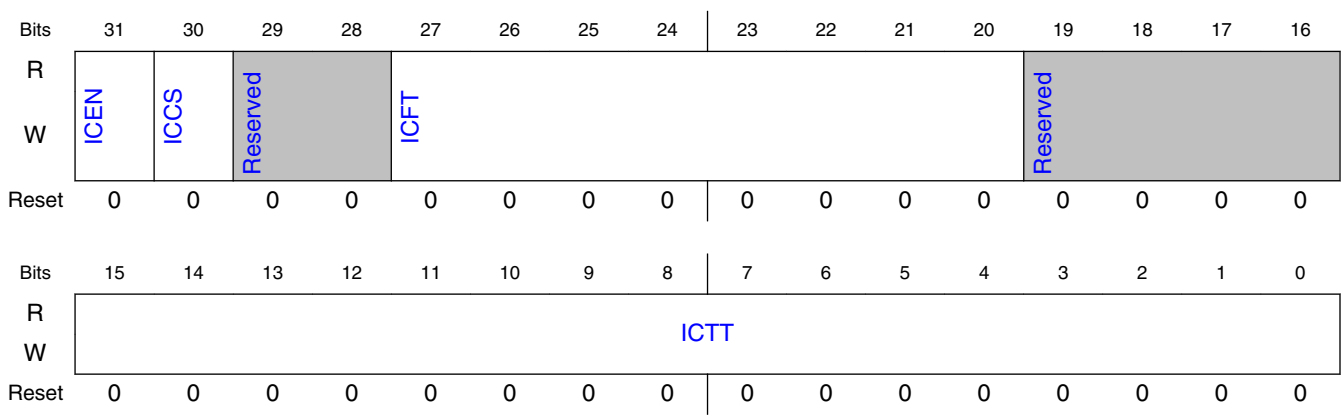
41.5.1.16.1 Offset

Register	Offset
RXIC0	100h

41.5.1.16.2 Function

See Interrupt coalescence for more information.

41.5.1.16.3 Diagram



41.5.1.16.4 Fields

Field	Function
31 ICEN	Interrupt Coalescing Enable 0b - Disable Interrupt coalescing. 1b - Enable Interrupt coalescing.
30	Interrupt Coalescing Timer Clock Source Select 0b - Use MII/GMII TX clocks.

Table continues on the next page...

Field	Function
ICCS	1b - Use ENET system clock.
29-28 —	This field must be set to 0.
27-20 ICFT	Interrupt coalescing frame count threshold This value determines the number of frames needed to be received for raising an interrupt. Frame counter restarts after reaching this threshold value or after the expiring of the coalescing timer. Must be greater than zero to avoid unpredictable behavior.
19-16 —	This field must be set to 0.
15-0 ICTT	Interrupt coalescing timer threshold Interrupt coalescing timer threshold in units of 64 clock periods. This value determines the maximum amount of time after receiving a frame before raising an interrupt. The threshold timer is disabled after expiring or number of frame reception defined by ICFT and starts again upon reception of the next first frame. Must be greater than zero to avoid unpredictable behavior.

41.5.1.17 Descriptor Individual Upper Address Register (IAUR)

41.5.1.17.1 Offset

Register	Offset
IAUR	118h

41.5.1.17.2 Function

IAUR contains the upper 32 bits of the 64-bit individual address hash table. The address recognition process uses this table to check for a possible match with the destination address (DA) field of receive frames with an individual DA. This register is not reset and you must initialize it.

41.5.1.17.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	IADDR1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IADDR1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

41.5.1.17.4 Fields

Field	Function
31-0 IADDR1	Contains the upper 32 bits of the 64-bit hash table used in the address recognition process for receive frames with a unicast address. Bit 31 of IADDR1 contains hash index bit 63. Bit 0 of IADDR1 contains hash index bit 32.

41.5.1.18 Descriptor Individual Lower Address Register (IALR)

41.5.1.18.1 Offset

Register	Offset
IALR	11Ch

41.5.1.18.2 Function

IALR contains the lower 32 bits of the 64-bit individual address hash table. The address recognition process uses this table to check for a possible match with the DA field of receive frames with an individual DA. This register is not reset and you must initialize it.

41.5.1.18.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	IADDR2															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IADDR2															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

41.5.1.18.4 Fields

Field	Function
31-0 IADDR2	Contains the lower 32 bits of the 64-bit hash table used in the address recognition process for receive frames with a unicast address. Bit 31 of IADDR2 contains hash index bit 31. Bit 0 of IADDR2 contains hash index bit 0.

41.5.1.19 Descriptor Group Upper Address Register (GAUR)

41.5.1.19.1 Offset

Register	Offset
GAUR	120h

41.5.1.19.2 Function

GAUR contains the upper 32 bits of the 64-bit hash table used in the address recognition process for receive frames with a multicast address. You must initialize this register.

41.5.1.19.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	GADDR1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GADDR1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

41.5.1.19.4 Fields

Field	Function
31-0 GADDR1	Contains the upper 32 bits of the 64-bit hash table used in the address recognition process for receive frames with a multicast address. Bit 31 of GADDR1 contains hash index bit 63. Bit 0 of GADDR1 contains hash index bit 32.

41.5.1.20 Descriptor Group Lower Address Register (GALR)

41.5.1.20.1 Offset

Register	Offset
GALR	124h

41.5.1.20.2 Function

GALR contains the lower 32 bits of the 64-bit hash table used in the address recognition process for receive frames with a multicast address. You must initialize this register.

41.5.1.20.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	GADDR2															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GADDR2															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

41.5.1.20.4 Fields

Field	Function
31-0 GADDR2	Contains the lower 32 bits of the 64-bit hash table used in the address recognition process for receive frames with a multicast address. Bit 31 of GADDR2 contains hash index bit 31. Bit 0 of GADDR2 contains hash index bit 0.

41.5.1.21 Transmit FIFO Watermark Register (TFWR)

41.5.1.21.1 Offset

Register	Offset
TFWR	144h

41.5.1.21.2 Function

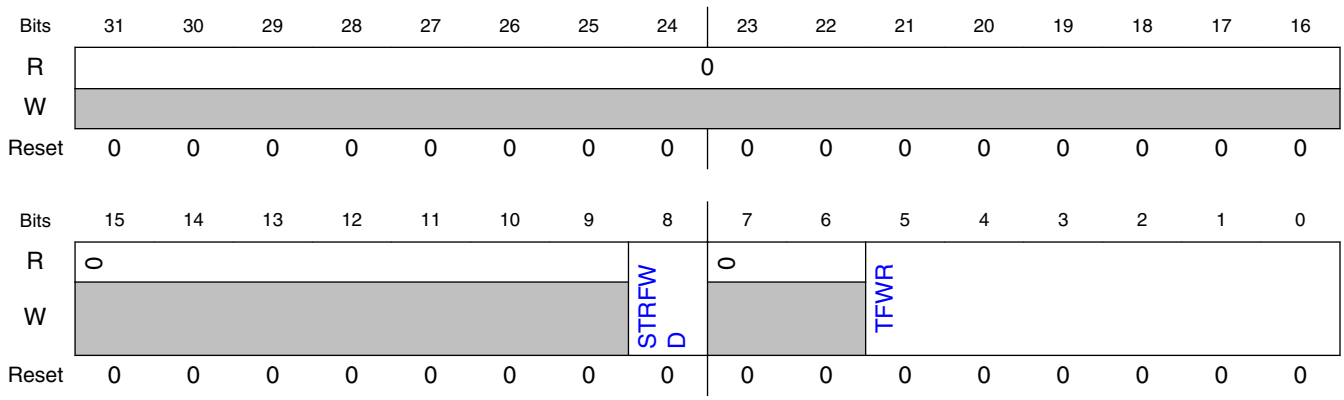
If TFWR[STRFWD] is cleared, TFWR[TFWR] controls the amount of data required in the transmit FIFO before transmission of a frame can begin. This allows you to minimize transmit latency (TFWR = 00 or 01) or allow for larger bus access latency (TFWR = 11) due to contention for the system bus. Setting the watermark to a high value minimizes the risk of transmit FIFO underrun due to contention for the system bus. The byte counts associated with the TFWR field may need to be modified to match a given system requirement, for example, worst-case bus access latency by the transmit data uDMA channel.

When the FIFO level reaches the value the TFWR field and when the STR_FWD is set to ‘0’, the MAC transmit control logic starts frame transmission even before the end-of-frame is available in the FIFO (cut-through operation).

If a complete frame has a size smaller than the threshold programmed with TFWR, the MAC also transmits the Frame to the line.

To enable store and forward on the Transmit path, set STR_FWD to ‘1’. In this case, the MAC starts to transmit data only when a complete frame is stored in the Transmit FIFO.

41.5.1.21.3 Diagram



41.5.1.21.4 Fields

Field	Function
31-9 —	Reserved
8 STRFWD	Store And Forward Enable 0b - Reset. The transmission start threshold is programmed in TFWR[TFWR]. 1b - Enabled.
7-6 —	Reserved
5-0 TFWR	Transmit FIFO Write If TFWR[STRFWD] is cleared, this field indicates the number of bytes, in steps of 64 bytes, written to the transmit FIFO before transmission of a frame begins. **NOTE:** If a frame with less than the threshold is written, it is still sent independently of this threshold setting. The threshold is relevant only if the frame is larger than the threshold given. 000000b - 64 bytes written. 000001b - 64 bytes written. 000010b - 128 bytes written. 000011b - 192 bytes written. 011111b - 1984 bytes written.

41.5.1.22 Receive Descriptor Ring 0 Start Register (RDSR)

41.5.1.22.1 Offset

Register	Offset
RDSR	180h

41.5.1.22.2 Function

RDSR points to the beginning of the circular receive buffer descriptor queue in external memory. This pointer must be 64-bit aligned (bits 2–0 must be zero); however, for optimal performance the pointer should be 512-bit aligned, that is, evenly divisible by 64.

NOTE

This register must be initialized prior to operation

41.5.1.22.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	R_DES_START															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	R_DES_START														0	
W														0		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

41.5.1.22.4 Fields

Field	Function
31-3 R_DES_START	Pointer to the beginning of the receive buffer descriptor queue.
2 —	This write-only field is reserved. It must always be written with the value 0.
1-0 —	Reserved

41.5.1.23 Transmit Buffer Descriptor Ring 0 Start Register (TDSR)

41.5.1.23.1 Offset

Register	Offset
TDSR	184h

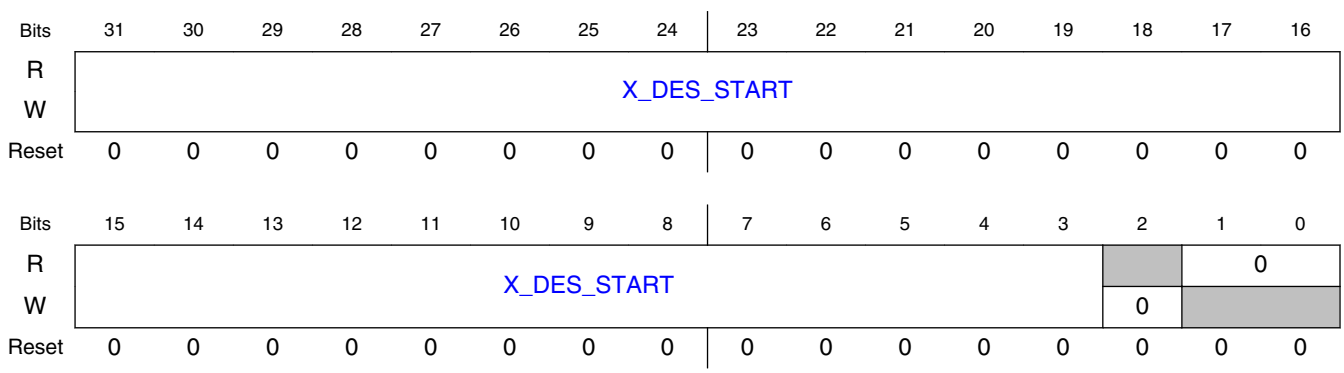
41.5.1.23.2 Function

TDSR provides a pointer to the beginning of the circular transmit buffer descriptor queue in external memory. This pointer must be 64-bit aligned (bits 2–0 must be zero); however, for optimal performance the pointer should be 512-bit aligned, that is, evenly divisible by 64.

NOTE

This register must be initialized prior to operation.

41.5.1.23.3 Diagram



41.5.1.23.4 Fields

Field	Function
31-3 X_DES_START	Pointer to the beginning of the transmit buffer descriptor queue.
2 —	This write-only field is reserved. It must always be written with the value 0.
1-0 —	Reserved

41.5.1.24 Maximum Receive Buffer Size Register - Ring 0 (MRBR)

41.5.1.24.1 Offset

Register	Offset
MRBR	188h

41.5.1.24.2 Function

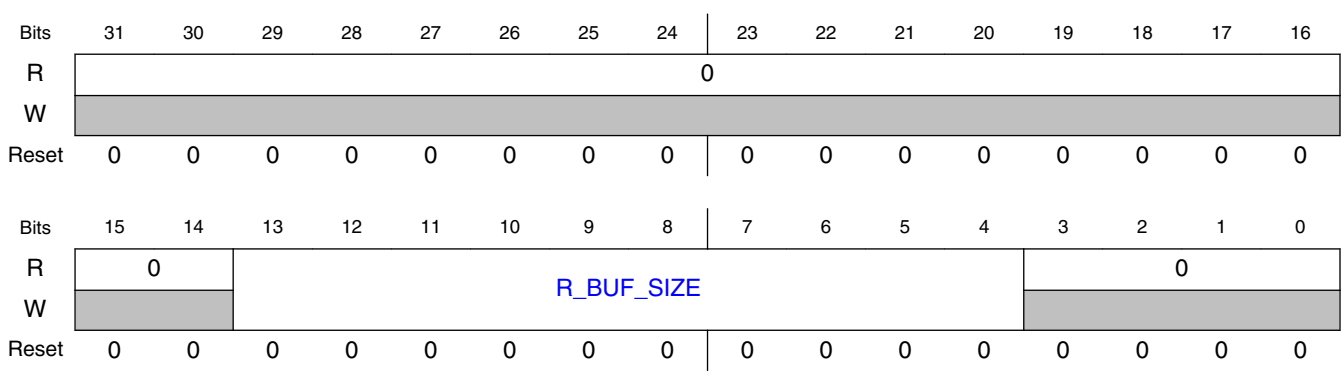
The MRBR is a user-programmable register that dictates the maximum size of all receive buffers. This value should take into consideration that the receive CRC is always written into the last receive buffer.

- R_BUF_SIZE is concatentated with the four least-significant bits of this register and are used as the maximum receive buffer size.
- To allow one maximum size frame per buffer, MRBR must be set to RCR[MAX_FL] or larger.
- To properly align the buffer, MRBR must be evenly divisible by 64. To ensure this, set the lower two bits of R_BUF_SIZE to zero. The lower four bits of this register are already set to zero by the device.
- To minimize bus usage (descriptor fetches), set MRBR greater than or equal to 256 bytes.

NOTE

This register must be initialized before operation.

41.5.1.24.3 Diagram



41.5.1.24.4 Fields

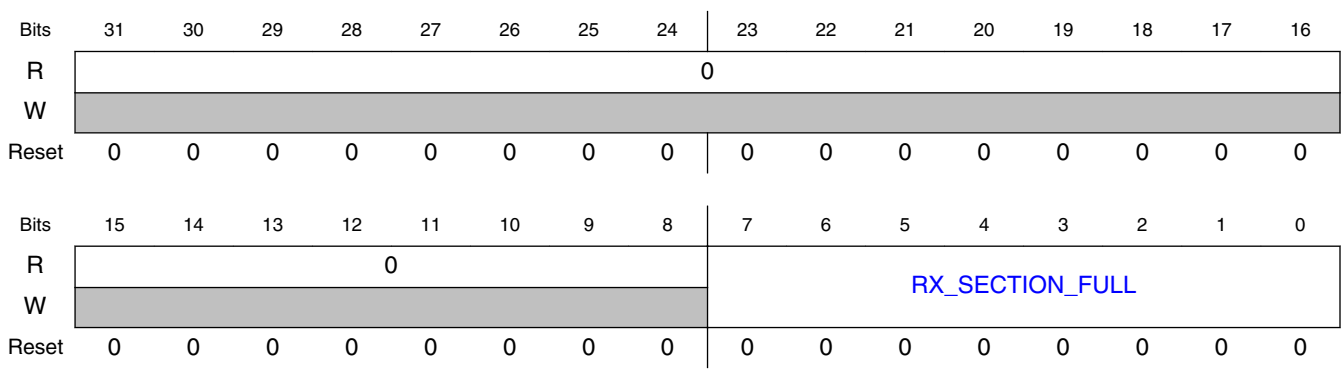
Field	Function
31-14 —	Reserved
13-4 R_BUF_SIZE	Receive buffer size in bytes. This value, concatenated with the four least-significant bits of this register (which are always zero), is the effective maximum receive buffer size.
3-0 —	This field, which is always zero, is the four least-significant bits of the maximum receive buffer size.

41.5.1.25 Receive FIFO Section Full Threshold (RSFL)

41.5.1.25.1 Offset

Register	Offset
RSFL	190h

41.5.1.25.2 Diagram



41.5.1.25.3 Fields

Field	Function
31-8 —	Reserved
7-0	Value Of Receive FIFO Section Full Threshold

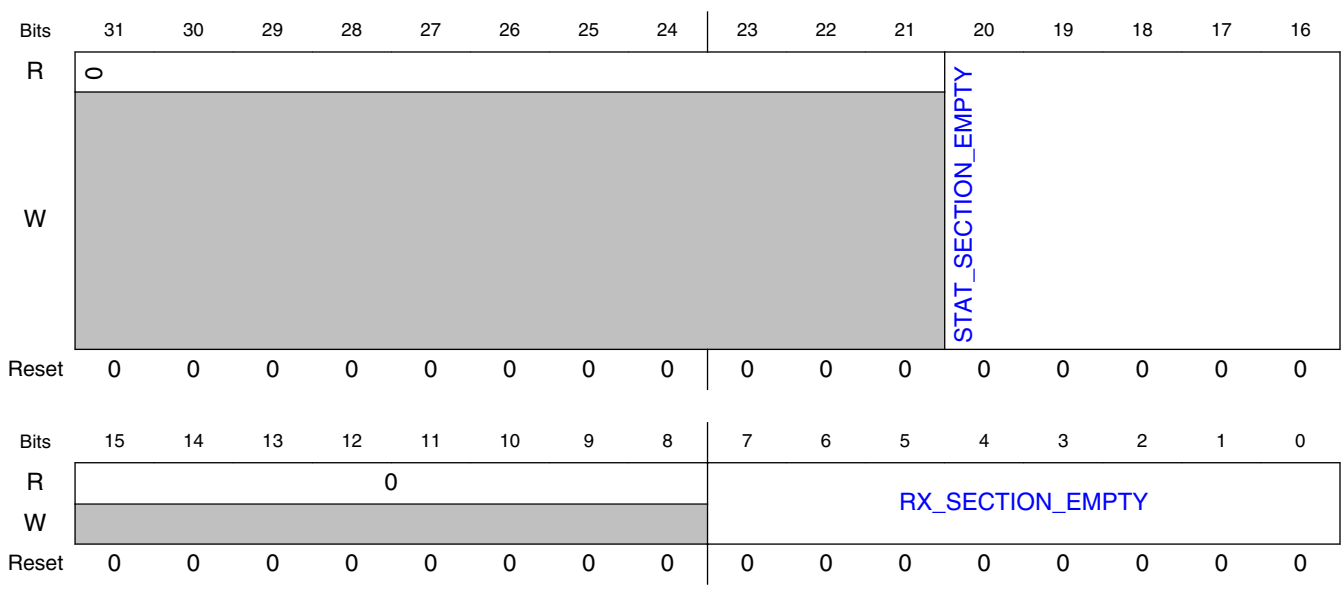
Field	Function
RX_SECTION_FULL	Value, in 64-bit words, of the receive FIFO section full threshold. Clear this field to enable store and forward on the RX FIFO. When programming a value greater than 0 (cut-through operation), it must be greater than RAEM[RX_ALMOST_EMPTY]. When the FIFO level reaches the value in this field, data is available in the Receive FIFO (cut-through operation).

41.5.1.26 Receive FIFO Section Empty Threshold (RSEM)

41.5.1.26.1 Offset

Register	Offset
RSEM	194h

41.5.1.26.2 Diagram



41.5.1.26.3 Fields

Field	Function
31-21 —	Reserved
20-16	RX Status FIFO Section Empty Threshold

Table continues on the next page...

Field	Function
STAT_SECTION_EMPTY	Defines number of frames in the receive FIFO, independent of its size, that can be accepted. If the limit is reached, reception will continue normally, however a pause frame will be triggered to indicate a possible congestion to the remote device to avoid FIFO overflow. A value of 0 disables automatic pause frame generation
15-8 —	Reserved
7-0 RX_SECTION_EMPTY	Value Of The Receive FIFO Section Empty Threshold Value, in 64-bit words, of the receive FIFO section empty threshold. When the FIFO has reached this level, a pause frame will be issued. A value of 0 disables automatic pause frame generation. When the FIFO level goes below the value programmed in this field, an XON pause frame is issued to indicate the FIFO congestion is cleared to the remote Ethernet client. **NOTE:** The section-empty threshold indications from both FIFOs are OR'ed to cause XOFF pause frame generation.

41.5.1.27 Receive FIFO Almost Empty Threshold (RAEM)

41.5.1.27.1 Offset

Register	Offset
RAEM	198h

41.5.1.27.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								RX_ALMOST_EMPTY							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

41.5.1.27.3 Fields

Field	Function
31-8	Reserved

Table continues on the next page...

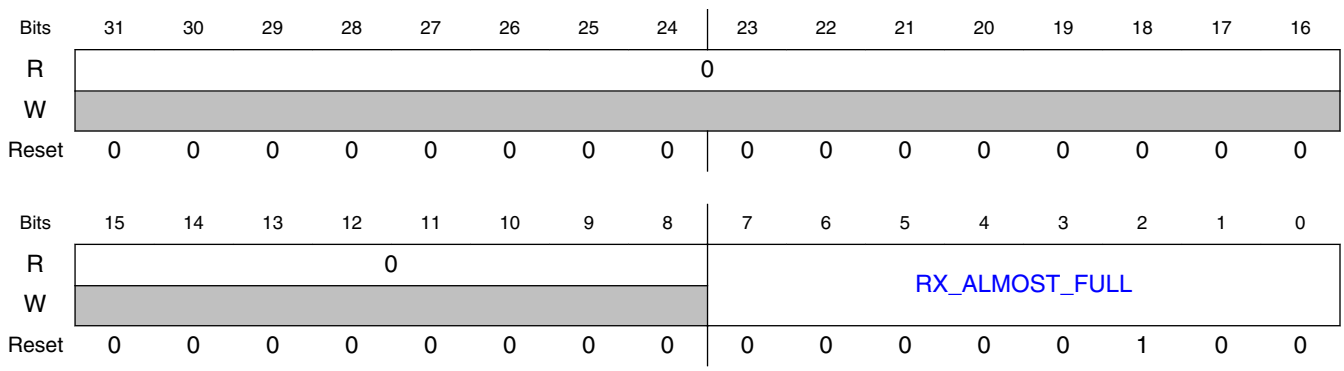
Field	Function
—	
7-0 RX_ALMOST_EMPTY	Value Of The Receive FIFO Almost Empty Threshold Value, in 64-bit words, of the receive FIFO almost empty threshold. When the FIFO level reaches the value programmed in this field and the end-of-frame has not been received for the frame yet, the core receive read control stops FIFO read (and subsequently stops transferring data to the MAC client application). It continues to deliver the frame, if again more data than the threshold or the end-of-frame is available in the FIFO. A minimum value of 4 should be set.

41.5.1.28 Receive FIFO Almost Full Threshold (RAFL)

41.5.1.28.1 Offset

Register	Offset
RAFL	19Ch

41.5.1.28.2 Diagram



41.5.1.28.3 Fields

Field	Function
31-8 —	Reserved
7-0 RX_ALMOST_FULL	Value Of The Receive FIFO Almost Full Threshold Value, in 64-bit words, of the receive FIFO almost full threshold. When the FIFO level comes close to the maximum, so that there is no more space for at least RX_ALMOST_FULL number of words, the MAC stops writing data in the FIFO and truncates the received frame to avoid FIFO overflow. The corresponding error status will be set when the frame is delivered to the application. A minimum value of 4 should be set.

41.5.1.29 Transmit FIFO Section Empty Threshold (TSEM)

41.5.1.29.1 Offset

Register	Offset
TSEM	1A0h

41.5.1.29.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								TX_SECTION_EMPTY							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

41.5.1.29.3 Fields

Field	Function
31-8 —	Reserved
7-0 TX_SECTION_EMPTY	Value Of The Transmit FIFO Section Empty Threshold Value, in 64-bit words, of the transmit FIFO section empty threshold. See Transmit FIFO for more information.

41.5.1.30 Transmit FIFO Almost Empty Threshold (TAEM)

41.5.1.30.1 Offset

Register	Offset
TAEM	1A4h

41.5.1.30.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								TX_ALMOST_EMPTY							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

41.5.1.30.3 Fields

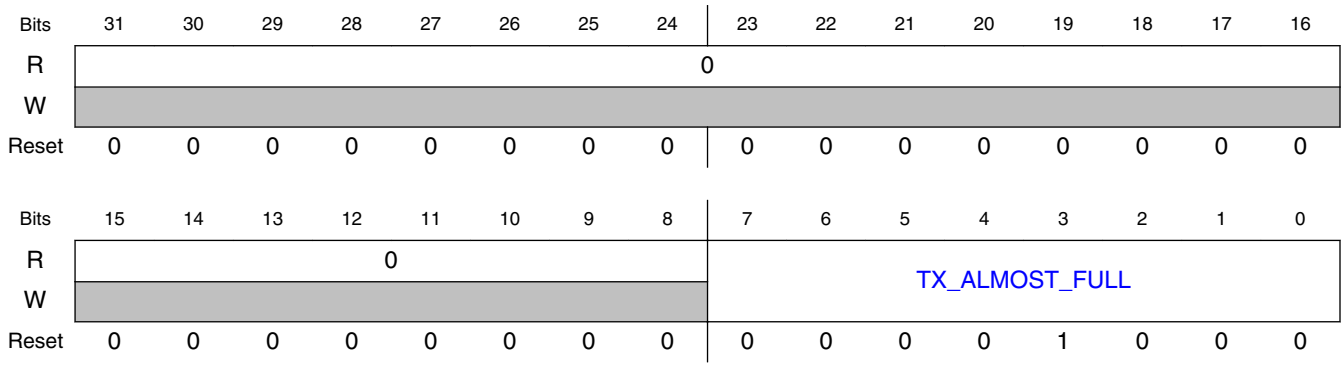
Field	Function
31-8 —	Reserved
7-0 TX_ALMOST_EMPTY	Value of Transmit FIFO Almost Empty Threshold Value, in 64-bit words, of the transmit FIFO almost empty threshold. When the FIFO level reaches the value programmed in this field, and no end-of-frame is available for the frame, the MAC transmit logic, to avoid FIFO underflow, stops reading the FIFO and transmits a frame with an MII error indication. See Transmit FIFO for more information. A minimum value of 4 should be set.

41.5.1.31 Transmit FIFO Almost Full Threshold (TAFL)

41.5.1.31.1 Offset

Register	Offset
TAFL	1A8h

41.5.1.31.2 Diagram



41.5.1.31.3 Fields

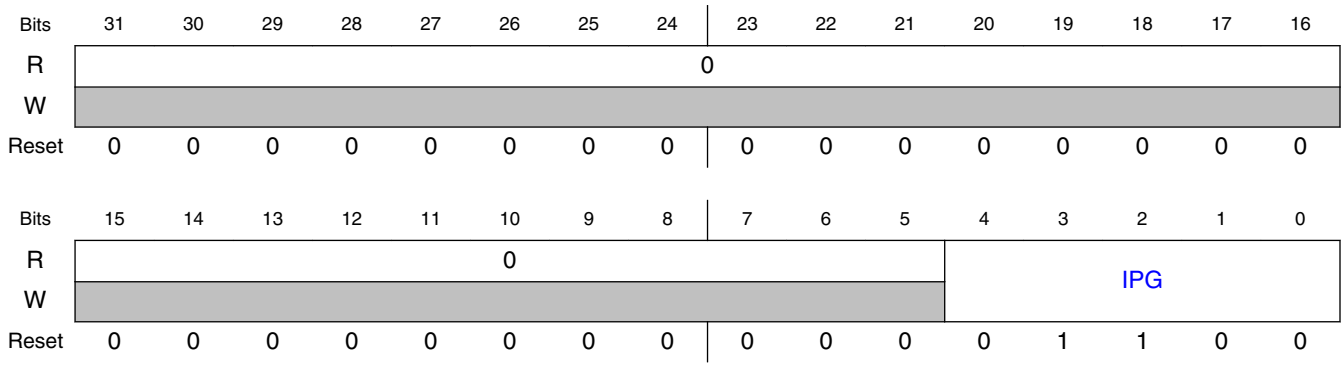
Field	Function
31-8 —	Reserved
7-0 TX_ALMOST_FULL	Value Of The Transmit FIFO Almost Full Threshold Value, in 64-bit words, of the transmit FIFO almost full threshold. A minimum value of six is required . A recommended value of at least 8 should be set allowing a latency of two clock cycles to the application. If more latency is required the value can be increased as necessary (latency = TAFL - 5). When the FIFO level comes close to the maximum, so that there is no more space for at least TX_ALMOST_FULL number of words, the pin ff_tx_rdy is deasserted. If the application does not react on this signal, the FIFO write control logic, to avoid FIFO overflow, truncates the current frame and sets the error status. As a result, the frame will be transmitted with an GMII/MII error indication. See Transmit FIFO for more information. **NOTE:** A FIFO overflow is a fatal error and requires a global reset on the transmit datapath or at least deassertion of ETHEREN.

41.5.1.32 Transmit Inter-Packet Gap (TIPG)

41.5.1.32.1 Offset

Register	Offset
TIPG	1ACh

41.5.1.32.2 Diagram



41.5.1.32.3 Fields

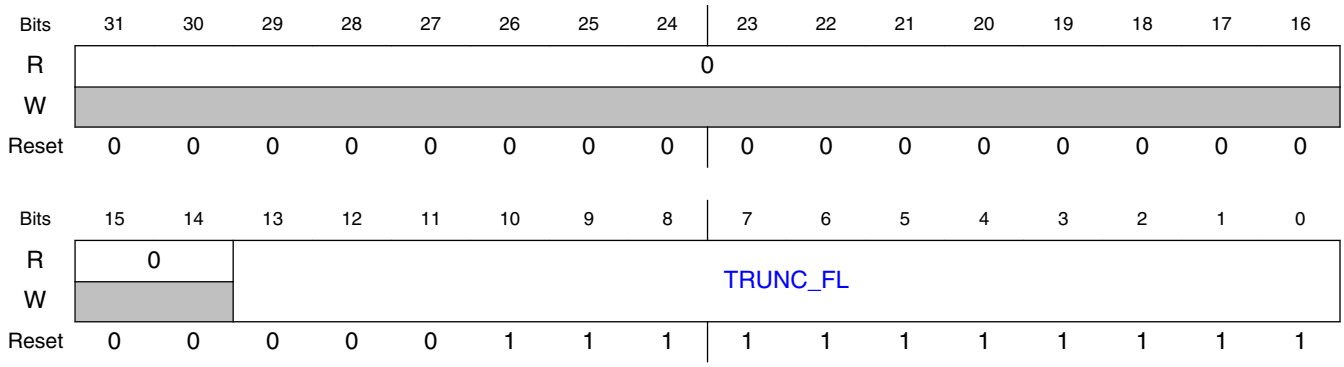
Field	Function
31-5 —	Reserved
4-0 IPG	Transmit Inter-Packet Gap Indicates the IPG, in bytes, between transmitted frames. Valid values range from 8 to 26. If the written value is less than 8 or greater than 26, the internal (effective) IPG is 12. **NOTE:** The IPG value read will be the value that was written, even if it is out of range.

41.5.1.33 Frame Truncation Length (FTRL)

41.5.1.33.1 Offset

Register	Offset
FTRL	1B0h

41.5.1.33.2 Diagram



41.5.1.33.3 Fields

Field	Function
31-14 —	Reserved
13-0 TRUNC_FL	Frame Truncation Length Indicates the value a receive frame is truncated, if it is greater than this value. Must be greater than or equal to RCR[MAX_FL]. **NOTE:** Truncation happens at TRUNC_FL. However, when truncation occurs, the application (FIFO) may receive less data, guaranteeing that it never receives more than the set limit.

41.5.1.34 Transmit Accelerator Function Configuration (TACC)

41.5.1.34.1 Offset

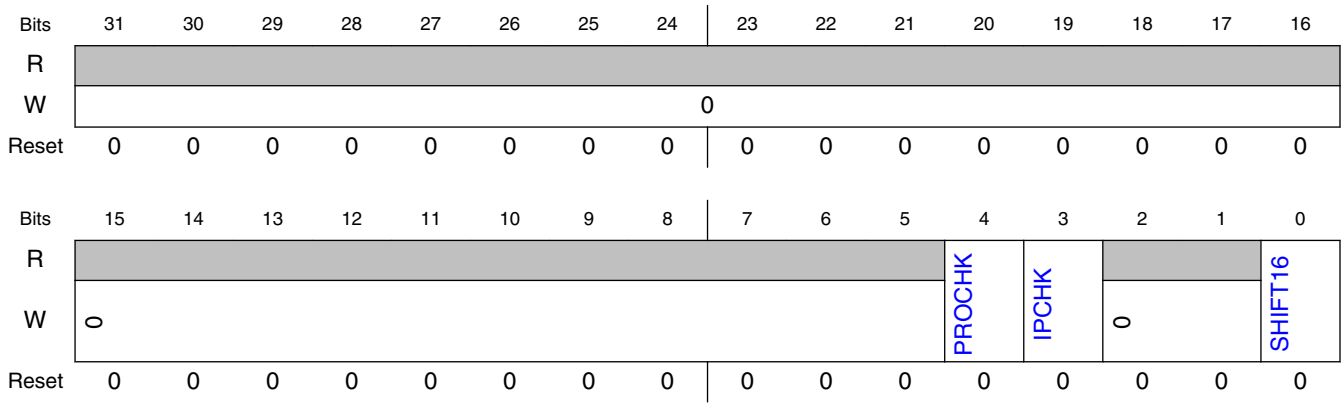
Register	Offset
TACC	1C0h

41.5.1.34.2 Function

TACC controls accelerator actions when sending frames. The register can be changed before or after each frame, but it must remain unmodified during frame writes into the transmit FIFO.

The TFWR[STRFWD] field must be set to use the checksum feature.

41.5.1.34.3 Diagram



41.5.1.34.4 Fields

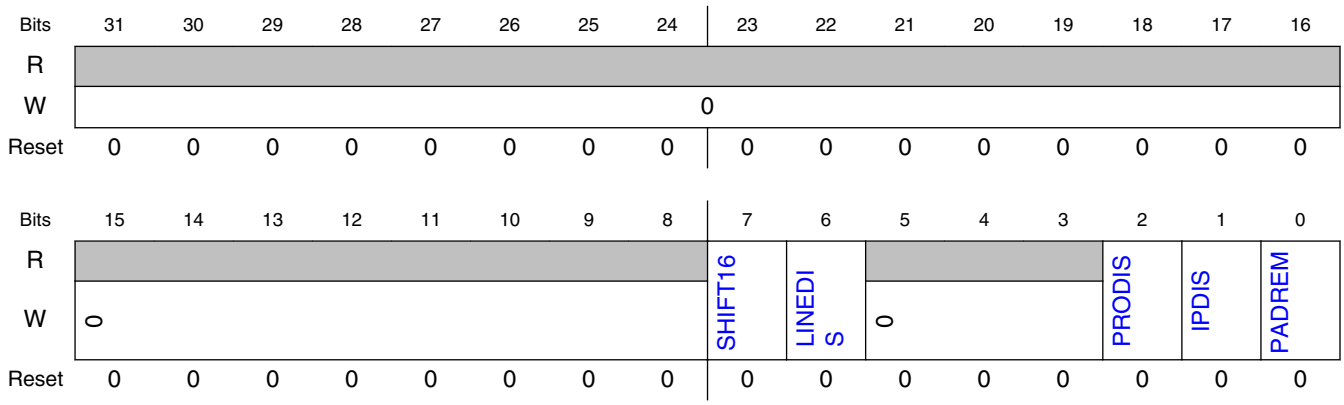
Field	Function
31-5 —	This write-only field is reserved. It must always be written with the value 0.
4 PROCHK	Enables insertion of protocol checksum. 0b - Checksum not inserted. 1b - If an IP frame with a known protocol is transmitted, the checksum is inserted automatically into the frame. The checksum field must be cleared. The other frames are not modified.
3 IPCHK	Enables insertion of IP header checksum. 0b - Checksum is not inserted. 1b - If an IP frame is transmitted, the checksum is inserted automatically. The IP header checksum field must be cleared. If a non-IP frame is transmitted the frame is not modified.
2-1 —	This write-only field is reserved. It must always be written with the value 0.
0 SHIFT16	TX FIFO Shift-16 0b - Disabled. 1b - Indicates to the transmit data FIFO that the written frames contain two additional octets before the frame data. This means the actual frame begins at bit 16 of the first word written into the FIFO. This function allows putting the frame payload on a 32-bit boundary in memory, as the 14-byte Ethernet header is extended to a 16-byte header.

41.5.1.35 Receive Accelerator Function Configuration (RACC)

41.5.1.35.1 Offset

Register	Offset
RACC	1C4h

41.5.1.35.2 Diagram



41.5.1.35.3 Fields

Field	Function
31-8 —	This write-only field is reserved. It must always be written with the value 0.
7 SHIFT16	RX FIFO Shift-16 When this field is set, the actual frame data starts at bit 16 of the first word read from the RX FIFO aligning the Ethernet payload on a 32-bit boundary. **NOTE:** This function only affects the FIFO storage and has no influence on the statistics, which use the actual length of the frame received. 0b - Disabled. 1b - Instructs the MAC to write two additional bytes in front of each frame received into the RX FIFO.
6 LINEDIS	Enable Discard Of Frames With MAC Layer Errors 0b - Frames with errors are not discarded. 1b - Any frame received with a CRC, length, or PHY error is automatically discarded and not forwarded to the user application interface.
5-3 —	This write-only field is reserved. It must always be written with the value 0.
2 PRODIS	Enable Discard Of Frames With Wrong Protocol Checksum 0b - Frames with wrong checksum are not discarded. 1b - If a TCP/IP, UDP/IP, or ICMP/IP frame is received that has a wrong TCP, UDP, or ICMP checksum, the frame is discarded. Discarding is only available when the RX FIFO operates in store and forward mode (RSFL cleared).
1 IPDIS	Enable Discard Of Frames With Wrong IPv4 Header Checksum 0b - Frames with wrong IPv4 header checksum are not discarded. 1b - If an IPv4 frame is received with a mismatching header checksum, the frame is discarded. IPv6 has no header checksum and is not affected by this setting. Discarding is only available when the RX FIFO operates in store and forward mode (RSFL cleared).
0 PADREM	Enable Padding Removal For Short IP Frames 0b - Padding not removed. 1b - Any bytes following the IP payload section of the frame are removed from the frame.

41.5.1.36 Tx Packet Count Statistic Register (RMON_T_PACKETS)

41.5.1.36.1 Offset

Register	Offset
RMON_T_PACKETS	204h

41.5.1.36.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXPKTS															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

41.5.1.36.3 Fields

Field	Function
31-16 —	Reserved
15-0 TXPKTS	Packet count Transmit packet count

41.5.1.37 Tx Broadcast Packets Statistic Register (RMON_T_BC_PKT)

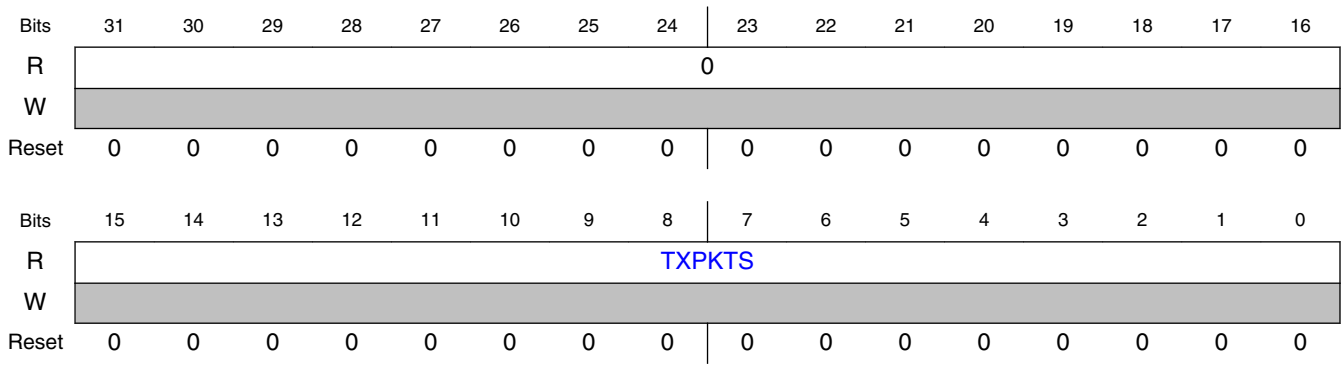
41.5.1.37.1 Offset

Register	Offset
RMON_T_BC_PKT	208h

41.5.1.37.2 Function

RMON Tx Broadcast Packets

41.5.1.37.3 Diagram



41.5.1.37.4 Fields

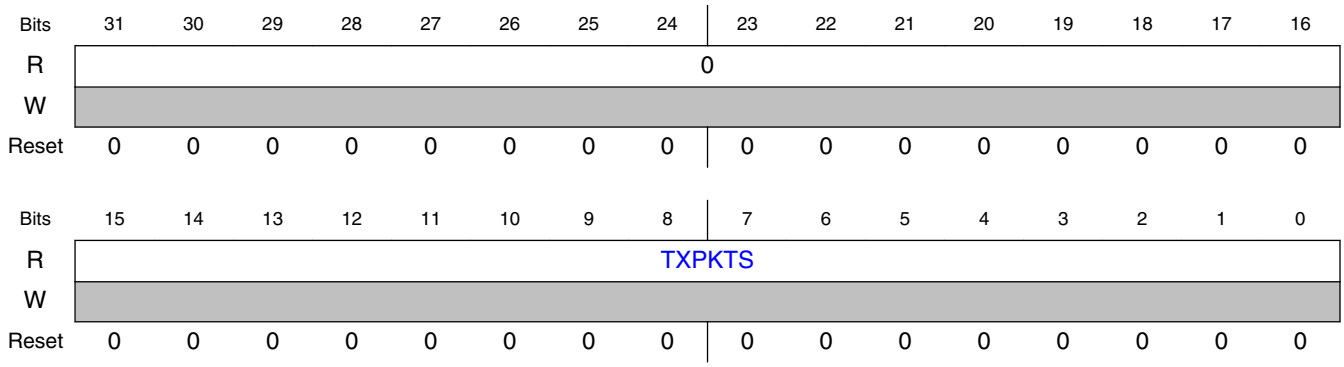
Field	Function
31-16 —	Reserved
15-0 TXPKTS	Number of broadcast packets

41.5.1.38 Tx Multicast Packets Statistic Register (RMON_T_MC_PKT)

41.5.1.38.1 Offset

Register	Offset
RMON_T_MC_PKT	20Ch

41.5.1.38.2 Diagram



41.5.1.38.3 Fields

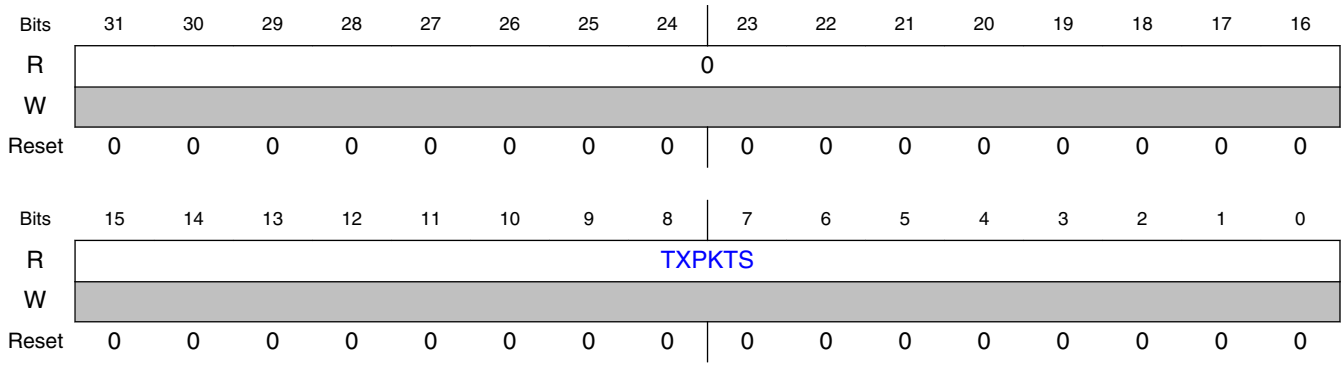
Field	Function
31-16 —	Reserved
15-0 TXPKTS	Number of multicast packets

41.5.1.39 Tx Packets with CRC/Align Error Statistic Register (RMON_T_CRC_ALIGN)

41.5.1.39.1 Offset

Register	Offset
RMON_T_CRC_ALIGN	210h

41.5.1.39.2 Diagram



41.5.1.39.3 Fields

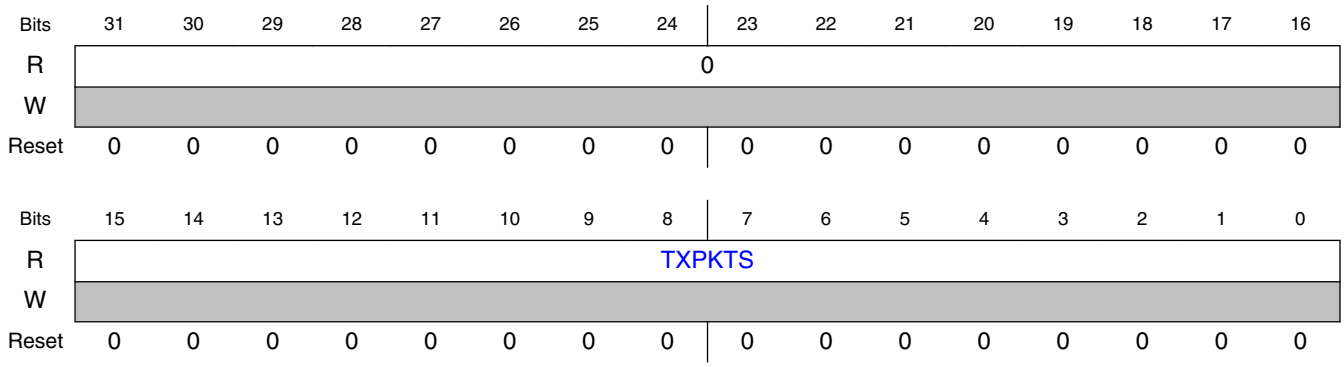
Field	Function
31-16 —	Reserved
15-0 TXPKTS	Number of packets with CRC/align error

41.5.1.40 Tx Packets Less Than Bytes and Good CRC Statistic Register (RMON_T_UNDERSIZE)

41.5.1.40.1 Offset

Register	Offset
RMON_T_UNDERSIZE	214h

41.5.1.40.2 Diagram



41.5.1.40.3 Fields

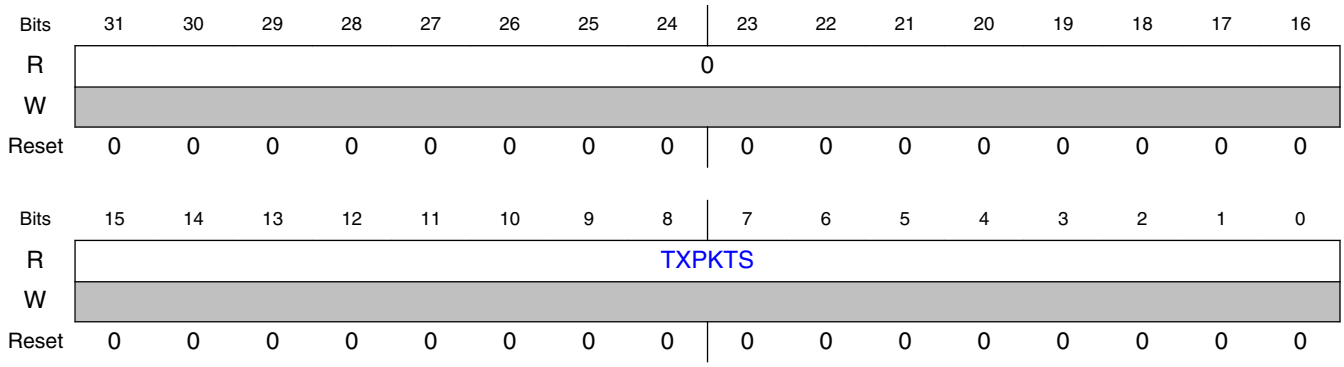
Field	Function
31-16 —	Reserved
15-0 TXPKTS	Number of transmit packets less than 64 bytes with good CRC

41.5.1.41 Tx Packets GT MAX_FL bytes and Good CRC Statistic Register (RMON_T_OVERSIZE)

41.5.1.41.1 Offset

Register	Offset
RMON_T_OVERSIZE	218h

41.5.1.41.2 Diagram



41.5.1.41.3 Fields

Field	Function
31-16 —	Reserved
15-0 TXPKTS	Number of transmit packets greater than MAX_FL bytes with good CRC

41.5.1.42 Tx Packets Less Than 64 Bytes and Bad CRC Statistic Register (RMON_T_FRAG)

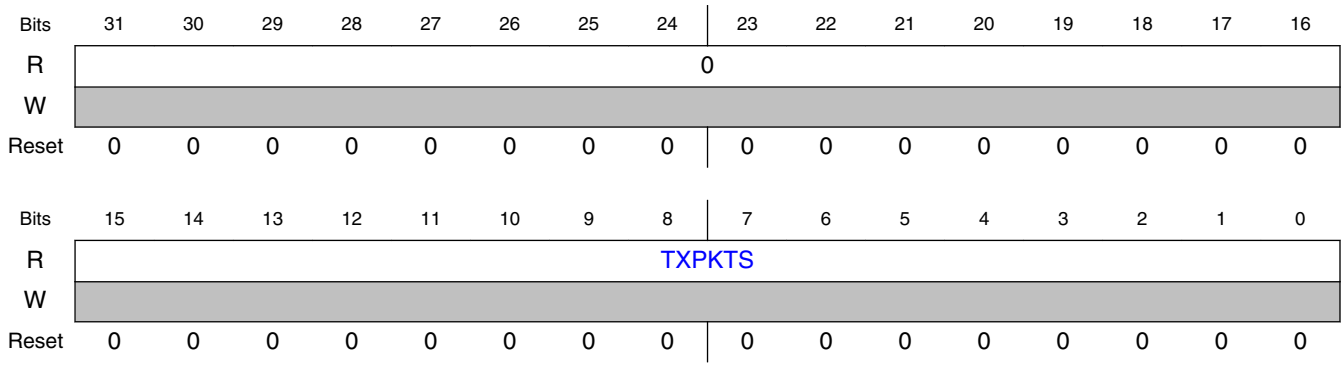
41.5.1.42.1 Offset

Register	Offset
RMON_T_FRAG	21Ch

41.5.1.42.2 Function

.

41.5.1.42.3 Diagram



41.5.1.42.4 Fields

Field	Function
31-16 —	Reserved
15-0 TXPKTS	Number of packets less than 64 bytes with bad CRC

41.5.1.43 Tx Packets Greater Than MAX_FL bytes and Bad CRC Statistic Register (RMON_T_JAB)

41.5.1.43.1 Offset

Register	Offset
RMON_T_JAB	220h

41.5.1.43.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXPKTS															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

41.5.1.43.3 Fields

Field	Function
31-16 —	Reserved
15-0 TXPKTS	Number of transmit packets greater than MAX_FL bytes and bad CRC

41.5.1.44 Tx Collision Count Statistic Register (RMON_T_COL)

41.5.1.44.1 Offset

Register	Offset
RMON_T_COL	224h

41.5.1.44.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXPKTS															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

41.5.1.44.3 Fields

Field	Function
31-16 —	Reserved
15-0 TXPKTS	Number of transmit collisions

41.5.1.45 Tx 64-Byte Packets Statistic Register (RMON_T_P64)

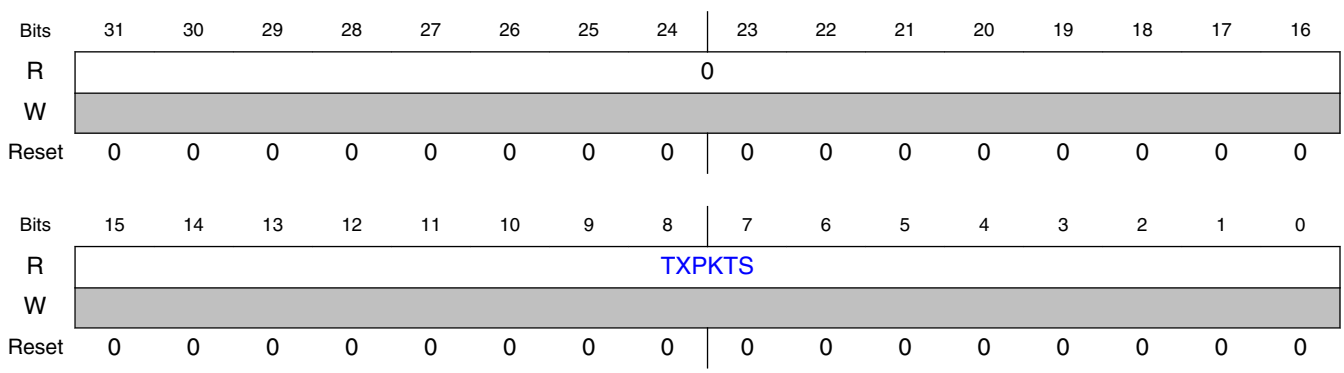
41.5.1.45.1 Offset

Register	Offset
RMON_T_P64	228h

41.5.1.45.2 Function

.

41.5.1.45.3 Diagram



41.5.1.45.4 Fields

Field	Function
31-16	Reserved

Table continues on the next page...

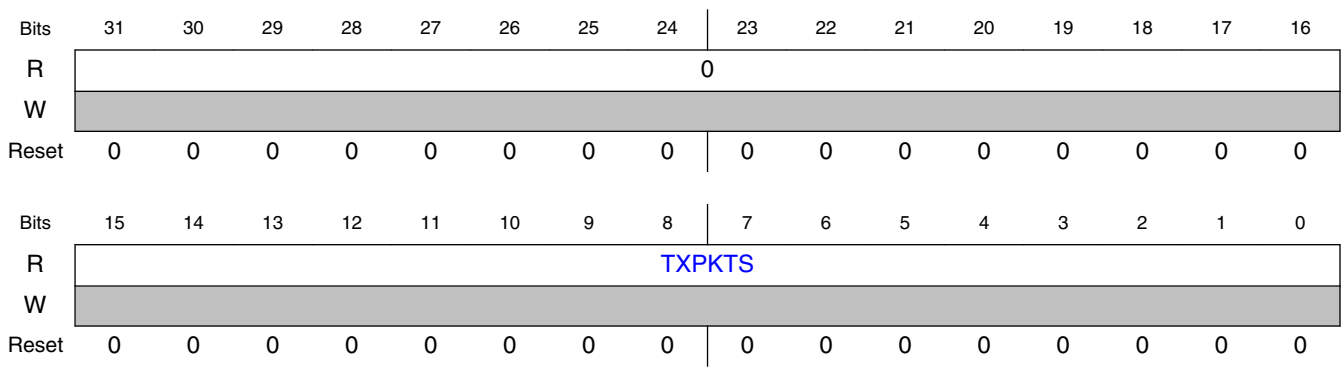
Field	Function
—	
15-0 TXPKTS	Number of 64-byte transmit packets

41.5.1.46 Tx 65- to 127-byte Packets Statistic Register (RMON_T_P65TO127)

41.5.1.46.1 Offset

Register	Offset
RMON_T_P65TO127	22Ch

41.5.1.46.2 Diagram



41.5.1.46.3 Fields

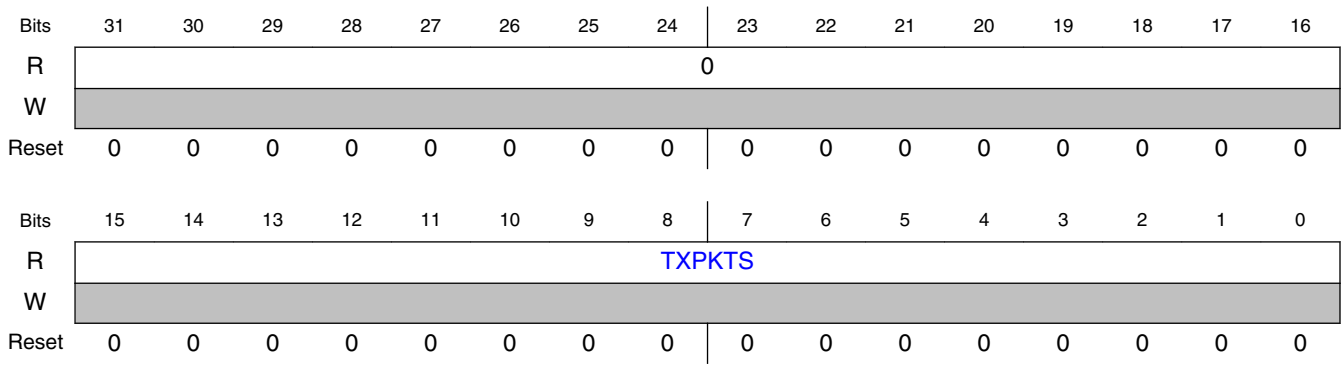
Field	Function
31-16 —	Reserved
15-0 TXPKTS	Number of 65- to 127-byte transmit packets

41.5.1.47 Tx 128- to 255-byte Packets Statistic Register (RMON_T_P128TO255)

41.5.1.47.1 Offset

Register	Offset
RMON_T_P128TO255	230h

41.5.1.47.2 Diagram



41.5.1.47.3 Fields

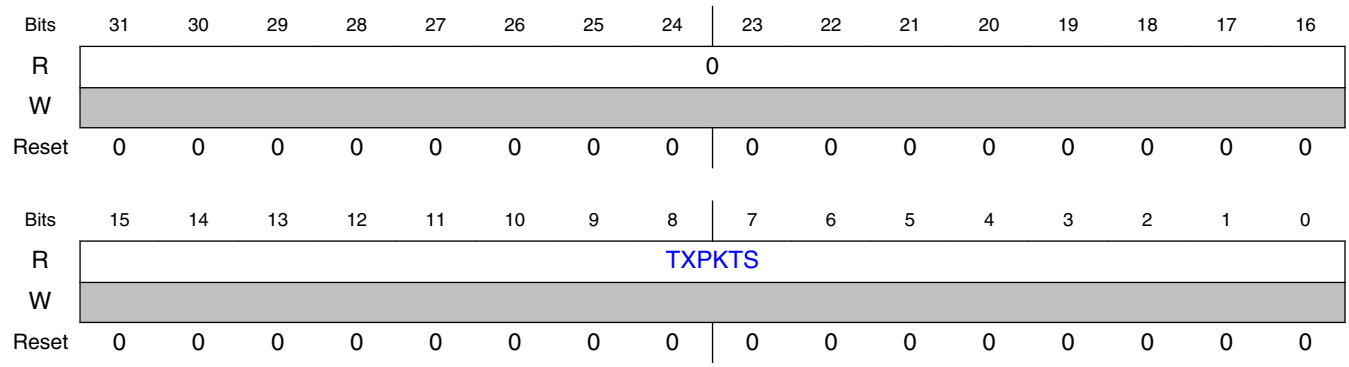
Field	Function
31-16 —	Reserved
15-0 TXPKTS	Number of 128- to 255-byte transmit packets

41.5.1.48 Tx 256- to 511-byte Packets Statistic Register (RMON_T_P256TO511)

41.5.1.48.1 Offset

Register	Offset
RMON_T_P256TO511	234h

41.5.1.48.2 Diagram



41.5.1.48.3 Fields

Field	Function
31-16 —	Reserved
15-0 TXPKTS	Number of 256- to 511-byte transmit packets

41.5.1.49 Tx 512- to 1023-byte Packets Statistic Register (RMON_T_P512TO1023)

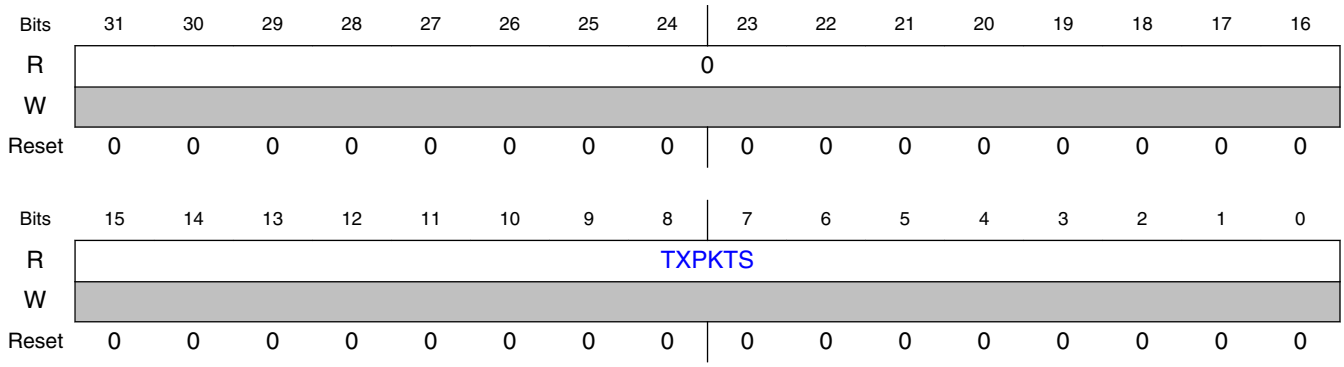
41.5.1.49.1 Offset

Register	Offset
RMON_T_P512TO1023	238h

41.5.1.49.2 Function

.

41.5.1.49.3 Diagram



41.5.1.49.4 Fields

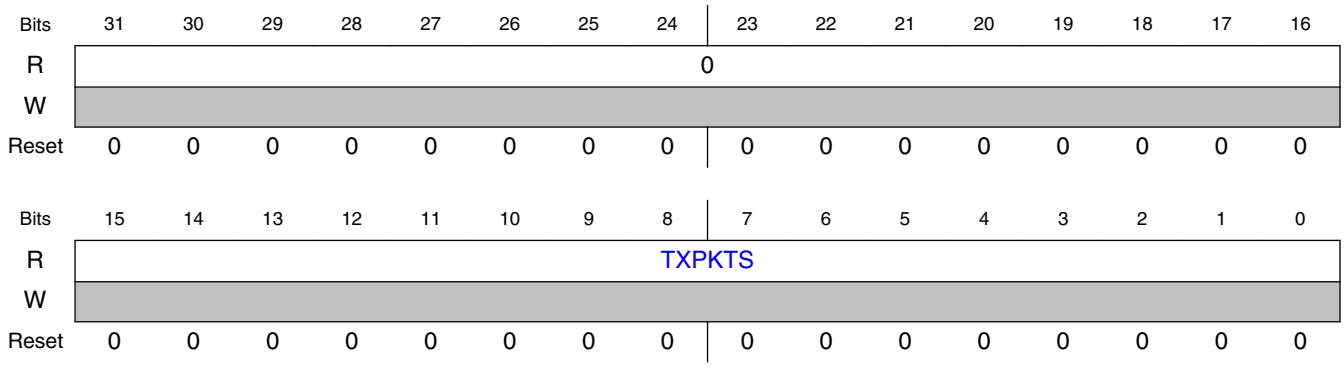
Field	Function
31-16 —	Reserved
15-0 TXPKTS	Number of 512- to 1023-byte transmit packets

41.5.1.50 Tx 1024- to 2047-byte Packets Statistic Register (RMON_T_P1024TO2047)

41.5.1.50.1 Offset

Register	Offset
RMON_T_P1024TO2047	23Ch

41.5.1.50.2 Diagram



41.5.1.50.3 Fields

Field	Function
31-16 —	Reserved
15-0 TXPKTS	Number of 1024- to 2047-byte transmit packets

41.5.1.51 Tx Packets Greater Than 2048 Bytes Statistic Register (RMON_T_P_GTE2048)

41.5.1.51.1 Offset

Register	Offset
RMON_T_P_GTE2048	240h

41.5.1.51.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXPKTS															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

41.5.1.51.3 Fields

Field	Function
31-16 —	Reserved
15-0 TXPKTS	Number of transmit packets greater than 2048 bytes

41.5.1.52 Tx Octets Statistic Register (RMON_T_OCTETS)

41.5.1.52.1 Offset

Register	Offset
RMON_T_OCTETS	244h

41.5.1.52.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TXOCTS															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXOCTS															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

41.5.1.52.3 Fields

Field	Function
31-0 TXOCTS	Number of transmit octets

41.5.1.53 Frames Transmitted OK Statistic Register (IEEE_T_FRAME_OK)

41.5.1.53.1 Offset

Register	Offset
IEEE_T_FRAME_OK	24Ch

41.5.1.53.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

41.5.1.53.3 Fields

Field	Function
31-16 —	Reserved
15-0 COUNT	Number of frames transmitted OK **NOTE:** Does not increment for the broadcast frames when broadcast reject is enabled and promiscuous mode is disabled within the receive control register (RCR).

41.5.1.54 Frames Transmitted with Single Collision Statistic Register (IEEE_T_1COL)

41.5.1.54.1 Offset

Register	Offset
IEEE_T_1COL	250h

41.5.1.54.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

41.5.1.54.3 Fields

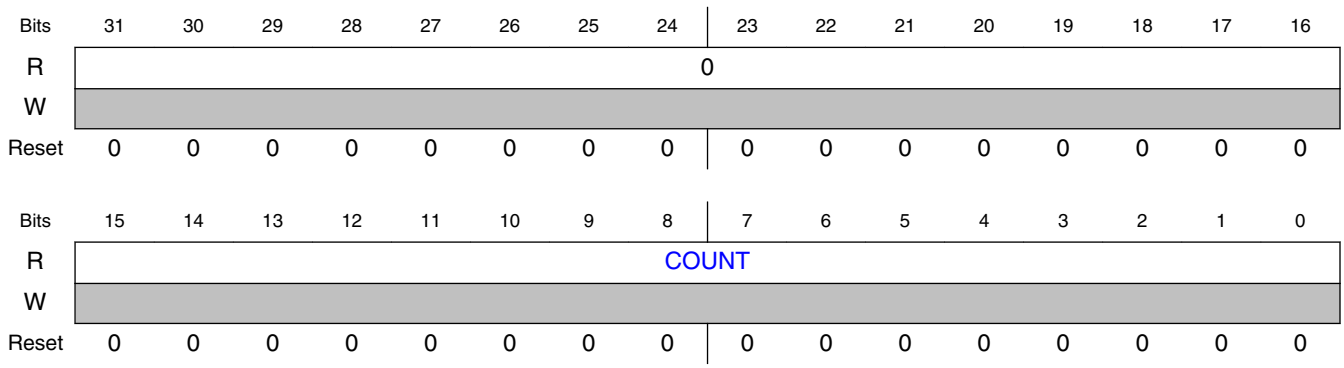
Field	Function
31-16 —	Reserved
15-0 COUNT	Number of frames transmitted with one collision

41.5.1.55 Frames Transmitted with Multiple Collisions Statistic Register (IEEE_T_MCOL)

41.5.1.55.1 Offset

Register	Offset
IEEE_T_MCOL	254h

41.5.1.55.2 Diagram



41.5.1.55.3 Fields

Field	Function
31-16 —	Reserved
15-0 COUNT	Number of frames transmitted with multiple collisions

41.5.1.56 Frames Transmitted after Deferral Delay Statistic Register (IEEE_T_DEF)

41.5.1.56.1 Offset

Register	Offset
IEEE_T_DEF	258h

41.5.1.56.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

41.5.1.56.3 Fields

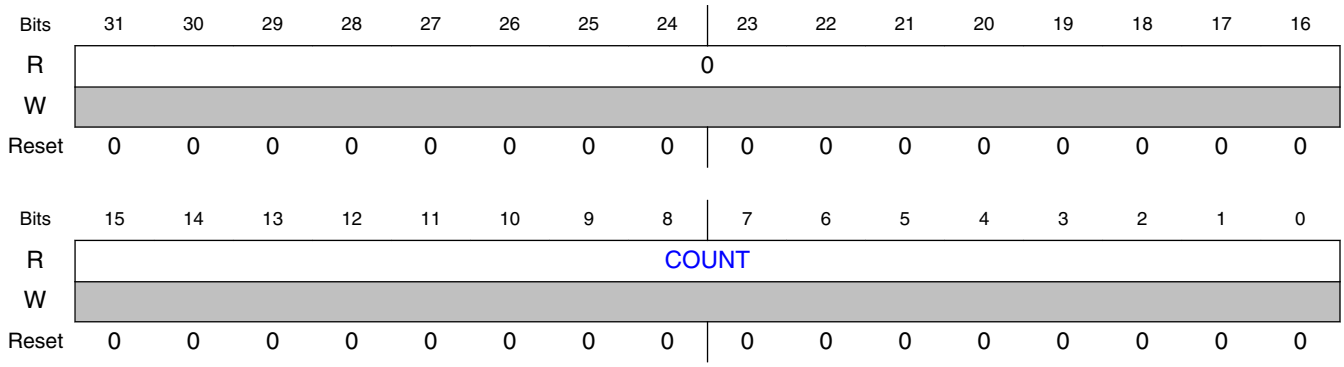
Field	Function
31-16 —	Reserved
15-0 COUNT	Number of frames transmitted with deferral delay

41.5.1.57 Frames Transmitted with Late Collision Statistic Register (IEEE_T_LCOL)

41.5.1.57.1 Offset

Register	Offset
IEEE_T_LCOL	25Ch

41.5.1.57.2 Diagram



41.5.1.57.3 Fields

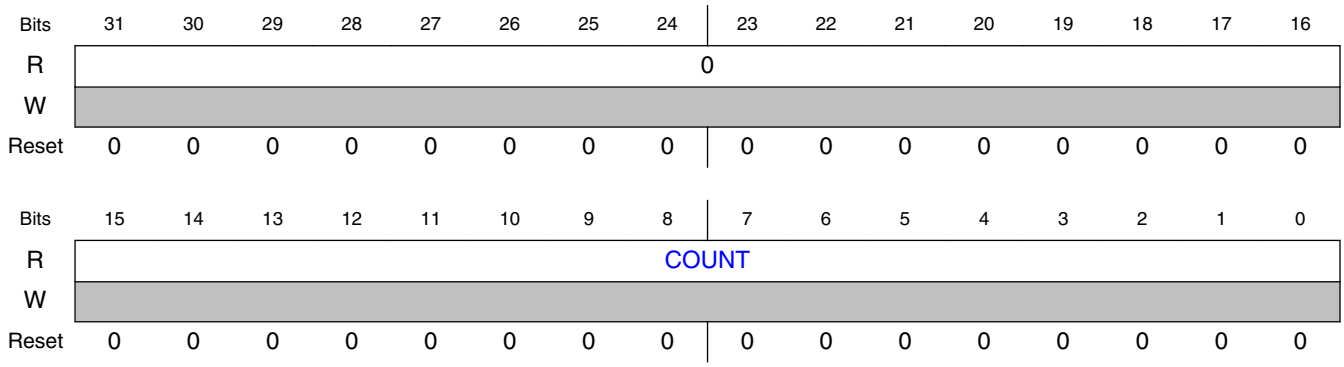
Field	Function
31-16 —	Reserved
15-0 COUNT	Number of frames transmitted with late collision

41.5.1.58 Frames Transmitted with Excessive Collisions Statistic Register (IEEE_T_EXCOL)

41.5.1.58.1 Offset

Register	Offset
IEEE_T_EXCOL	260h

41.5.1.58.2 Diagram



41.5.1.58.3 Fields

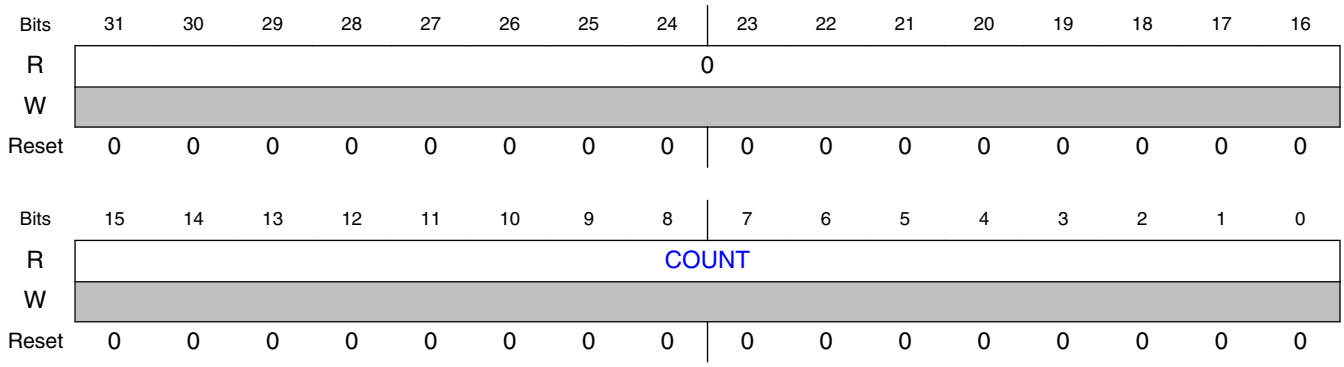
Field	Function
31-16 —	Reserved
15-0 COUNT	Number of frames transmitted with excessive collisions

41.5.1.59 Frames Transmitted with Tx FIFO Underrun Statistic Register (IEEE_T_MACERR)

41.5.1.59.1 Offset

Register	Offset
IEEE_T_MACERR	264h

41.5.1.59.2 Diagram



41.5.1.59.3 Fields

Field	Function
31-16 —	Reserved
15-0 COUNT	Number of frames transmitted with transmit FIFO underrun

41.5.1.60 Frames Transmitted with Carrier Sense Error Statistic Register (IEEE_T_CSERR)

41.5.1.60.1 Offset

Register	Offset
IEEE_T_CSERR	268h

41.5.1.60.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

41.5.1.60.3 Fields

Field	Function
31-16 —	Reserved
15-0 COUNT	Number of frames transmitted with carrier sense error

41.5.1.61 Reserved Statistic Register (IEEE_T_SQE)

41.5.1.61.1 Offset

Register	Offset
IEEE_T_SQE	26Ch

41.5.1.61.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

41.5.1.61.3 Fields

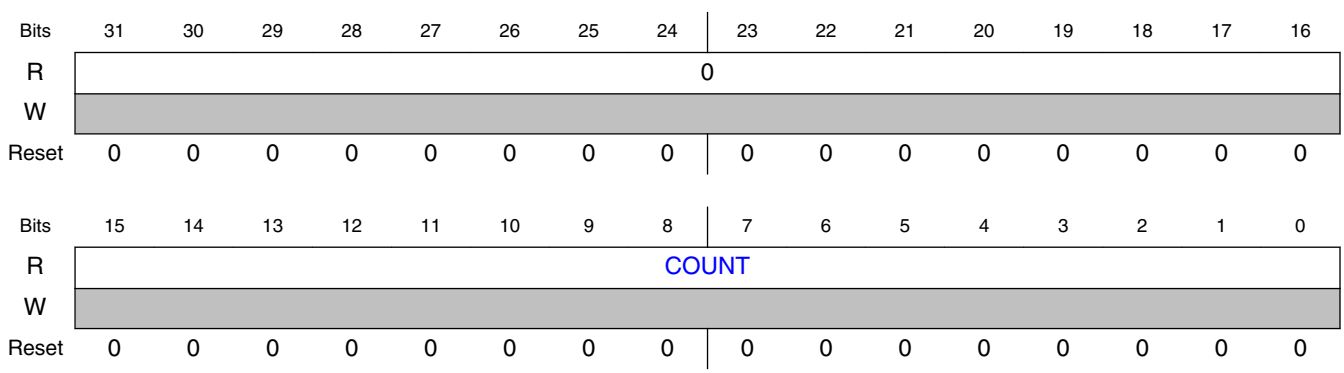
Field	Function
31-16 —	Reserved
15-0 COUNT	This read-only field is reserved and always has the value 0 This read-only field is reserved and always has the value 0. **NOTE:** Counter not implemented as no SQE information is available.

41.5.1.62 Flow Control Pause Frames Transmitted Statistic Register (IEEE_T_FDXFC)

41.5.1.62.1 Offset

Register	Offset
IEEE_T_FDXFC	270h

41.5.1.62.2 Diagram



41.5.1.62.3 Fields

Field	Function
31-16 —	Reserved

Table continues on the next page...

Field	Function
15-0 COUNT	Number of flow-control pause frames transmitted

41.5.1.63 Octet Count for Frames Transmitted w/o Error Statistic Register (IEEE_T_OCTETS_OK)

41.5.1.63.1 Offset

Register	Offset
IEEE_T_OCTETS_OK	274h

41.5.1.63.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

41.5.1.63.3 Fields

Field	Function
31-0 COUNT	Octet count for frames transmitted without error Counts total octets (includes header and FCS fields).

41.5.1.64 Rx Packet Count Statistic Register (RMON_R_PACKETS)

41.5.1.64.1 Offset

Register	Offset
RMON_R_PACKETS	284h

41.5.1.64.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

41.5.1.64.3 Fields

Field	Function
31-16 —	Reserved
15-0 COUNT	Number of packets received

41.5.1.65 Rx Broadcast Packets Statistic Register (RMON_R_BC_PKT)

41.5.1.65.1 Offset

Register	Offset
RMON_R_BC_PKT	288h

41.5.1.65.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

41.5.1.65.3 Fields

Field	Function
31-16 —	Reserved
15-0 COUNT	Number of receive broadcast packets

41.5.1.66 Rx Multicast Packets Statistic Register (RMON_R_MC_PKT)

41.5.1.66.1 Offset

Register	Offset
RMON_R_MC_PKT	28Ch

41.5.1.66.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

41.5.1.66.3 Fields

Field	Function
31-16 —	Reserved
15-0 COUNT	Number of receive multicast packets

41.5.1.67 Rx Packets with CRC/Align Error Statistic Register (RMON_R_CRC_ALIGN)

41.5.1.67.1 Offset

Register	Offset
RMON_R_CRC_ALIGN	290h

41.5.1.67.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

41.5.1.67.3 Fields

Field	Function
31-16 —	Reserved
15-0 COUNT	Number of receive packets with CRC or align error

41.5.1.68 Rx Packets with Less Than 64 Bytes and Good CRC Statistic Register (RMON_R_UNDERSIZE)

41.5.1.68.1 Offset

Register	Offset
RMON_R_UNDERSIZE	294h

41.5.1.68.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

41.5.1.68.3 Fields

Field	Function
31-16 —	Reserved
15-0 COUNT	Number of receive packets with less than 64 bytes and good CRC

41.5.1.69 Rx Packets Greater Than MAX_FL and Good CRC Statistic Register (RMON_R_OVERSIZE)

41.5.1.69.1 Offset

Register	Offset
RMON_R_OVERSIZE	298h

41.5.1.69.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

41.5.1.69.3 Fields

Field	Function
31-16 —	Reserved
15-0 COUNT	Number of receive packets greater than MAX_FL and good CRC

41.5.1.70 Rx Packets Less Than 64 Bytes and Bad CRC Statistic Register (RMON_R_FRAG)

41.5.1.70.1 Offset

Register	Offset
RMON_R_FRAG	29Ch

41.5.1.70.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

41.5.1.70.3 Fields

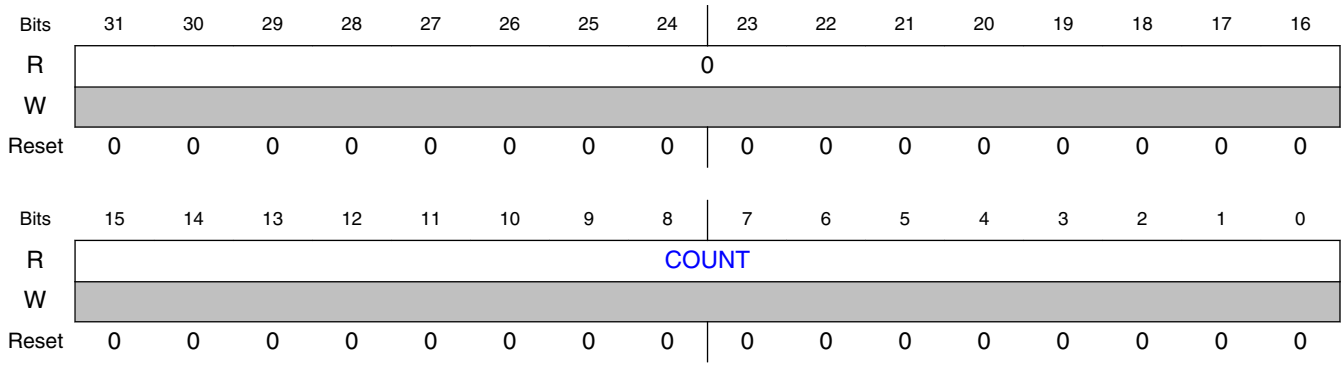
Field	Function
31-16 —	Reserved
15-0 COUNT	Number of receive packets with less than 64 bytes and bad CRC

41.5.1.71 Rx Packets Greater Than MAX_FL Bytes and Bad CRC Statistic Register (RMON_R_JAB)

41.5.1.71.1 Offset

Register	Offset
RMON_R_JAB	2A0h

41.5.1.71.2 Diagram



41.5.1.71.3 Fields

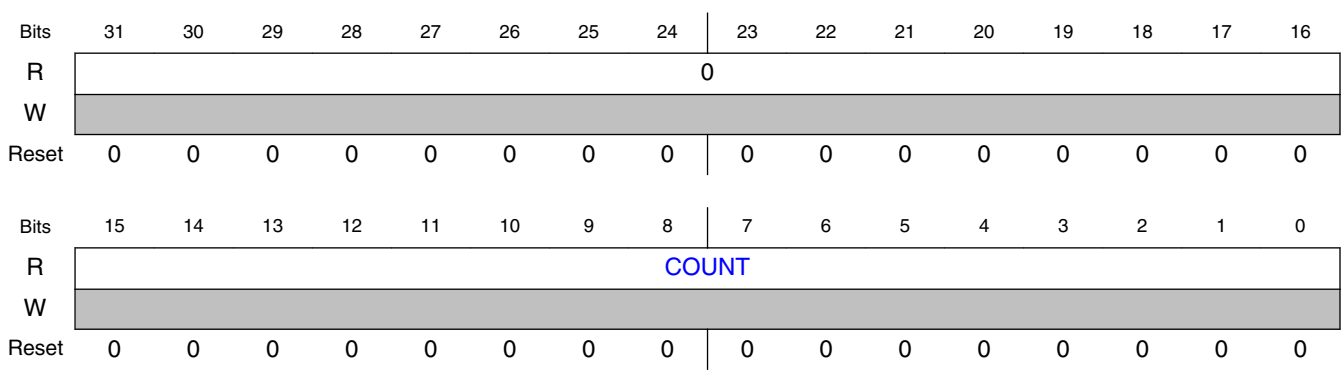
Field	Function
31-16 —	Reserved
15-0 COUNT	Number of receive packets greater than MAX_FL and bad CRC

41.5.1.72 Rx 64-Byte Packets Statistic Register (RMON_R_P64)

41.5.1.72.1 Offset

Register	Offset
RMON_R_P64	2A8h

41.5.1.72.2 Diagram



41.5.1.72.3 Fields

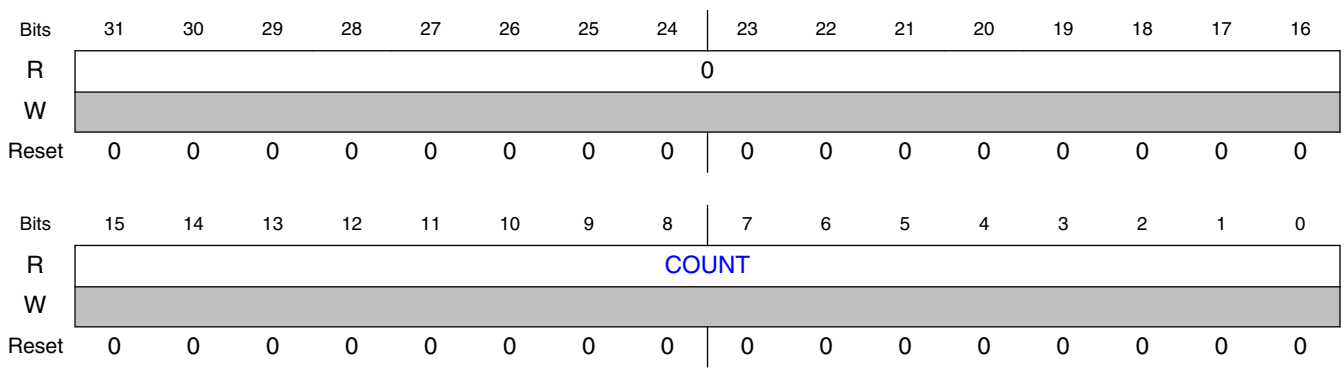
Field	Function
31-16 —	Reserved
15-0 COUNT	Number of 64-byte receive packets

41.5.1.73 Rx 65- to 127-Byte Packets Statistic Register (RMON_R_P65TO127)

41.5.1.73.1 Offset

Register	Offset
RMON_R_P65TO127	2ACh

41.5.1.73.2 Diagram



41.5.1.73.3 Fields

Field	Function
31-16 —	Reserved
15-0 COUNT	Number of 65- to 127-byte recieve packets

41.5.1.74 Rx 128- to 255-Byte Packets Statistic Register (RMON_R_P128TO255)

41.5.1.74.1 Offset

Register	Offset
RMON_R_P128TO255	2B0h

41.5.1.74.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

41.5.1.74.3 Fields

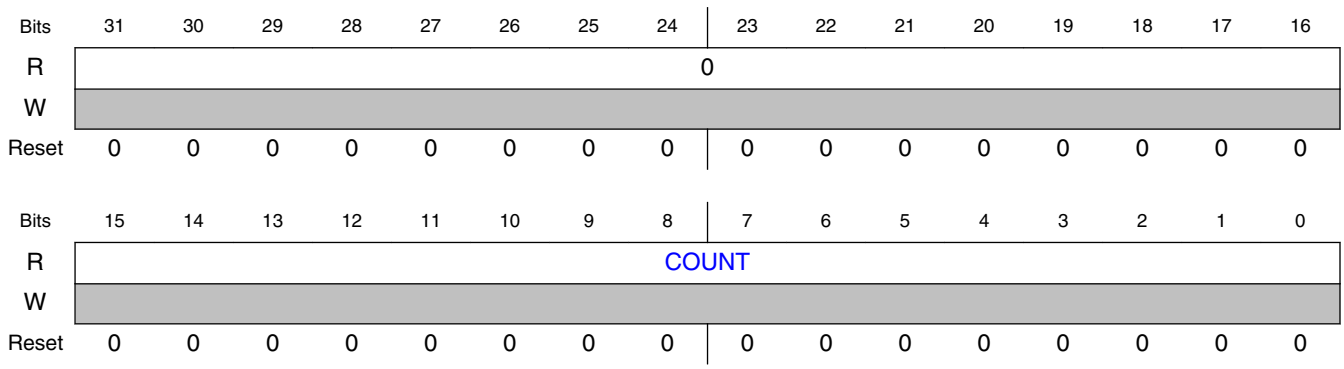
Field	Function
31-16 —	Reserved
15-0 COUNT	Number of 128- to 255-byte recieve packets

41.5.1.75 Rx 256- to 511-Byte Packets Statistic Register (RMON_R_P256TO511)

41.5.1.75.1 Offset

Register	Offset
RMON_R_P256TO511	2B4h

41.5.1.75.2 Diagram



41.5.1.75.3 Fields

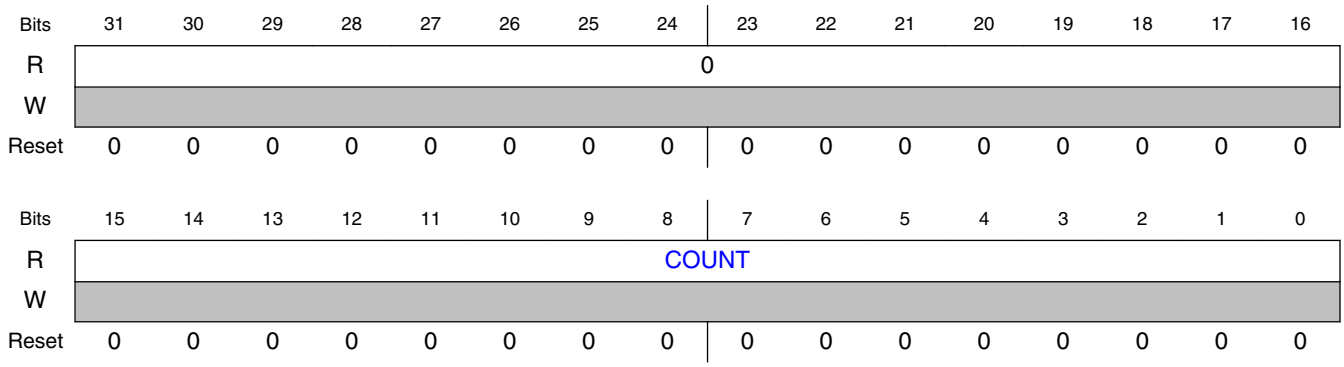
Field	Function
31-16 —	Reserved
15-0 COUNT	Number of 256- to 511-byte recieve packets

41.5.1.76 Rx 512- to 1023-Byte Packets Statistic Register (RMON_R_P512TO1023)

41.5.1.76.1 Offset

Register	Offset
RMON_R_P512TO1023	2B8h

41.5.1.76.2 Diagram



41.5.1.76.3 Fields

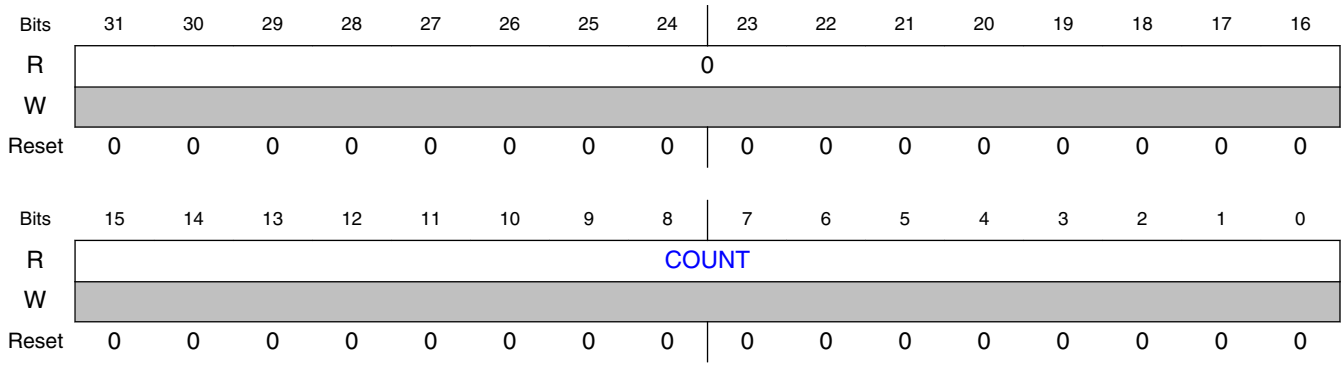
Field	Function
31-16 —	Reserved
15-0 COUNT	Number of 512- to 1023-byte recieve packets

41.5.1.77 Rx 1024- to 2047-Byte Packets Statistic Register (RMON_R_P1024TO2047)

41.5.1.77.1 Offset

Register	Offset
RMON_R_P1024TO2047	2BCh

41.5.1.77.2 Diagram



41.5.1.77.3 Fields

Field	Function
31-16 —	Reserved
15-0 COUNT	Number of 1024- to 2047-byte recieve packets

41.5.1.78 Rx Packets Greater than 2048 Bytes Statistic Register (RMON_R_P_GTE2048)

41.5.1.78.1 Offset

Register	Offset
RMON_R_P_GTE2048	2C0h

41.5.1.78.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

41.5.1.78.3 Fields

Field	Function
31-16 —	Reserved
15-0 COUNT	Number of greater-than-2048-byte recieve packets

41.5.1.79 Rx Octets Statistic Register (RMON_R_OCTETS)

41.5.1.79.1 Offset

Register	Offset
RMON_R_OCTETS	2C4h

41.5.1.79.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

41.5.1.79.3 Fields

Field	Function
31-0 COUNT	Number of receive octets

41.5.1.80 Frames not Counted Correctly Statistic Register (IEEE_R_DROP)

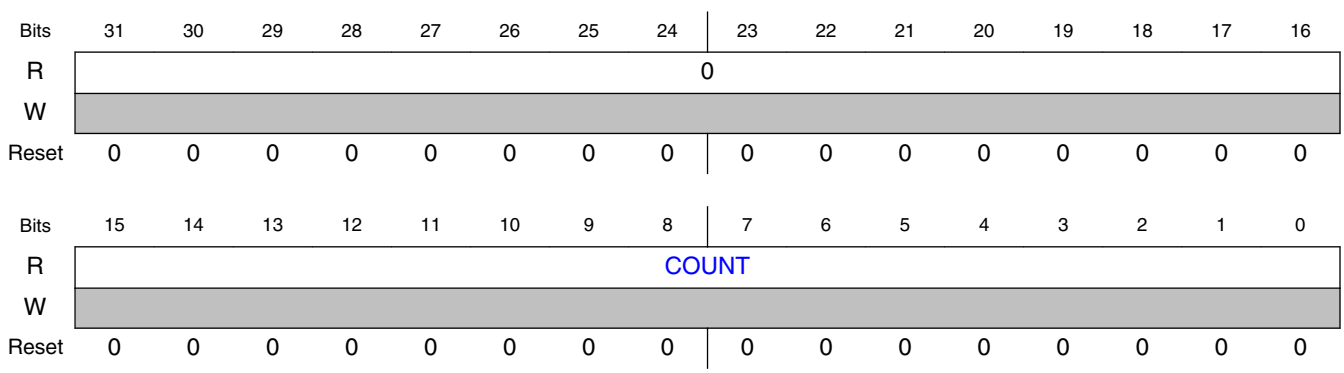
41.5.1.80.1 Offset

Register	Offset
IEEE_R_DROP	2C8h

41.5.1.80.2 Function

Counter increments if a frame with invalid or missing SFD character is detected and has been dropped. None of the other counters increments if this counter increments.

41.5.1.80.3 Diagram



41.5.1.80.4 Fields

Field	Function
31-16	Reserved

Table continues on the next page...

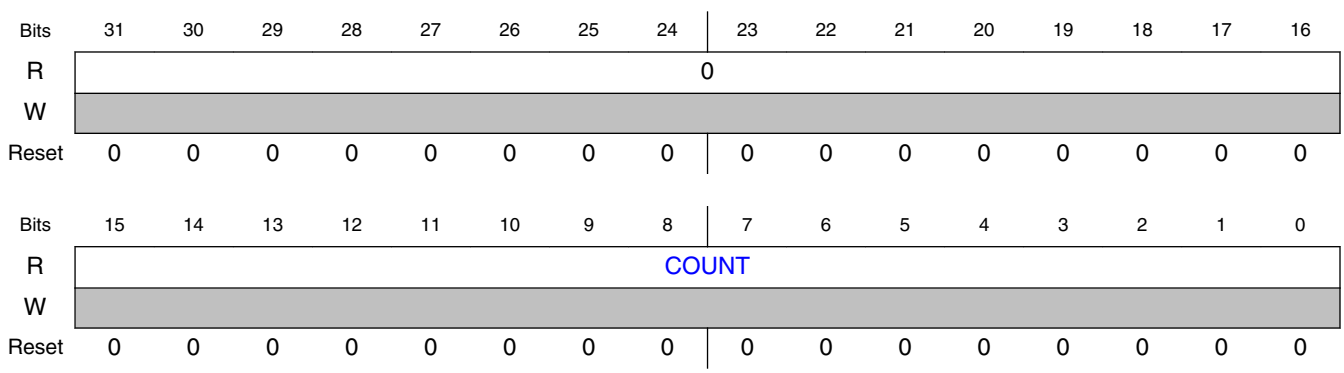
Field	Function
—	
15-0 COUNT	Frame count

41.5.1.81 Frames Received OK Statistic Register (IEEE_R_FRAME_OK)

41.5.1.81.1 Offset

Register	Offset
IEEE_R_FRAME_OK	2CCh

41.5.1.81.2 Diagram



41.5.1.81.3 Fields

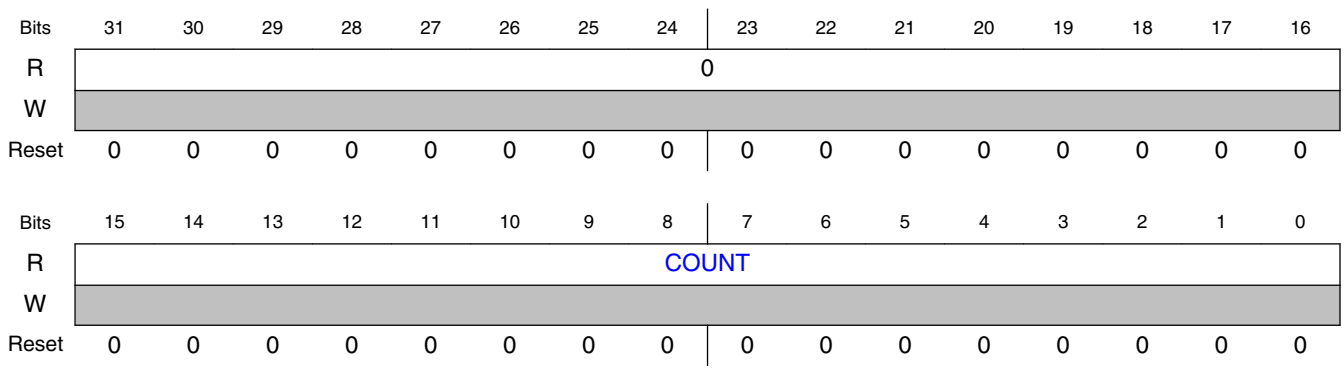
Field	Function
31-16 —	Reserved
15-0 COUNT	Number of frames received OK

41.5.1.82 Frames Received with CRC Error Statistic Register (IEEE_R_CRC)

41.5.1.82.1 Offset

Register	Offset
IEEE_R_CRC	2D0h

41.5.1.82.2 Diagram



41.5.1.82.3 Fields

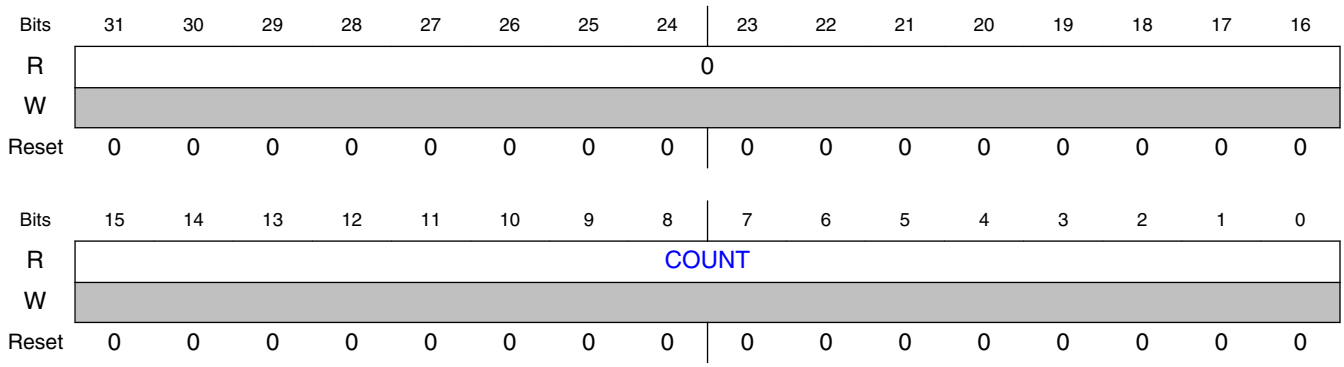
Field	Function
31-16 —	Reserved
15-0 COUNT	Number of frames received with CRC error

41.5.1.83 Frames Received with Alignment Error Statistic Register (IEEE_R_ALIGN)

41.5.1.83.1 Offset

Register	Offset
IEEE_R_ALIGN	2D4h

41.5.1.83.2 Diagram



41.5.1.83.3 Fields

Field	Function
31-16 —	Reserved
15-0 COUNT	Number of frames received with alignment error

41.5.1.84 Receive FIFO Overflow Count Statistic Register (IEEE_R_MACERR)

41.5.1.84.1 Offset

Register	Offset
IEEE_R_MACERR	2D8h

41.5.1.84.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

41.5.1.84.3 Fields

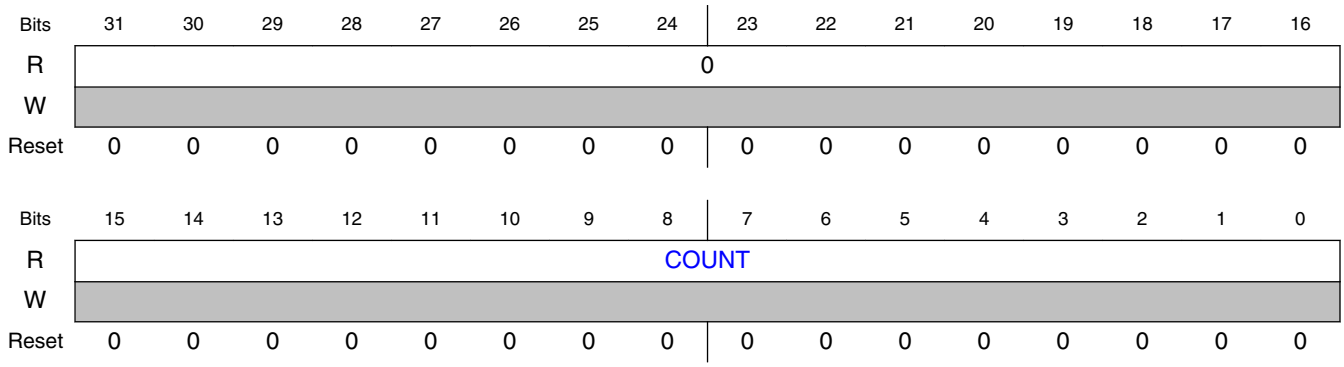
Field	Function
31-16 —	Reserved
15-0 COUNT	Receive FIFO overflow count

41.5.1.85 Flow Control Pause Frames Received Statistic Register (IEEE_R_FDXFC)

41.5.1.85.1 Offset

Register	Offset
IEEE_R_FDXFC	2DCh

41.5.1.85.2 Diagram



41.5.1.85.3 Fields

Field	Function
31-16 —	Reserved
15-0 COUNT	Number of flow-control pause frames received

41.5.1.86 Octet Count for Frames Received without Error Statistic Register (IEEE_R_OCTETS_OK)

41.5.1.86.1 Offset

Register	Offset
IEEE_R_OCTETS_OK	2E0h

41.5.1.86.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

41.5.1.86.3 Fields

Field	Function
31-0 COUNT	Number of octets for frames received without error **NOTE:** Counts total octets (includes header and FCS fields). Does not increment for the broadcast frames when broadcast reject is enabled and promiscuous mode is disabled within the receive control register (RCR).

41.5.1.87 Adjustable Timer Control Register (ATCR)

41.5.1.87.1 Offset

Register	Offset
ATCR	400h

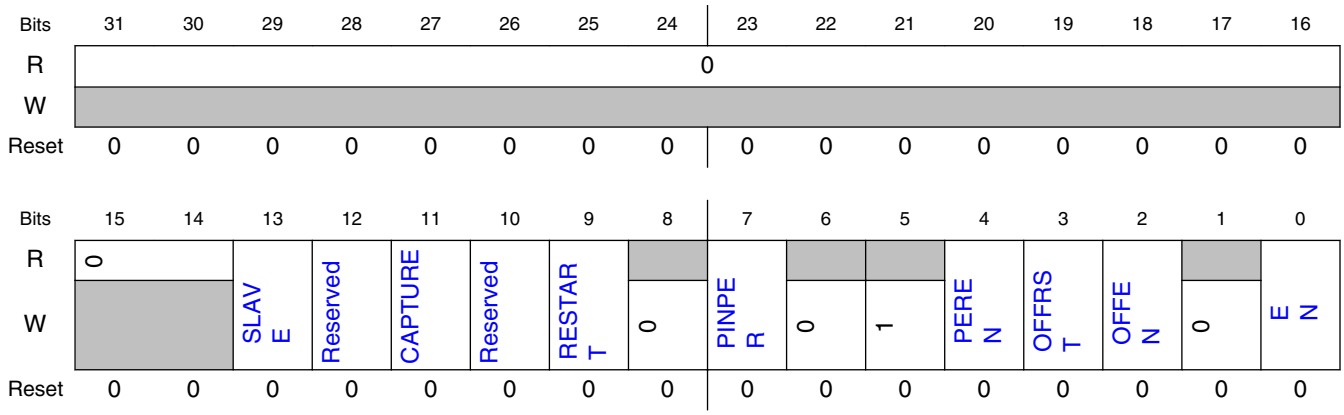
41.5.1.87.2 Function

ATCR command fields can trigger the corresponding events directly. It is not necessary to preserve any of the configuration fields when a command field is set in the register, that is, no read-modify-write is required.

NOTE

The CAPTURE and RESTART fields and bits 12 and 10 must be 0 in order to write to the other fields in this register.

41.5.1.87.3 Diagram



41.5.1.87.4 Fields

Field	Function
31-14 —	Reserved
13 SLAVE	Enable Timer Slave Mode 0b - The timer is active and all configuration fields in this register are relevant. 1b - The internal timer is disabled and the externally provided timer value is used. All other fields, except CAPTURE, in this register have no effect. CAPTURE can still be used to capture the current timer value.
12 —	Always write 0 to this field Always write 0 to this field.
11 CAPTURE	Capture Timer Value When this field is set, all other fields are ignored during a write. This field automatically clears to 0 after the command completes. 0b - No effect. 1b - The current time is captured and can be read from the ATVR register.
10 —	Always write 0 to this field Always write 0 to this field.
9 RESTART	Reset Timer Resets the timer to zero. This has no effect on the counter enable. If the counter is enabled when this field is set, the timer is reset to zero and starts counting from there. When set, all other fields are ignored during a write. This field automatically clears to 0 after the command completes. RESTART should be used when the timer is enabled.
8 —	Reserved
7 PINPER	Enables event signal output external pin frc_evt_period assertion on period event Enables event signal output assertion on period event. **NOTE:** Not all devices contain the event signal output. See the chip configuration details. 0b - Disable. 1b - Enable.

Table continues on the next page...

Field	Function
6 —	Reserved
5 —	This field must be written always with one **NOTE:** This field must be written always with one.
4 PEREN	Enable Periodical Event 0b - Disable. 1b - A period event interrupt can be generated (EIR[TS_TIMER]) and the event signal output is asserted when the timer wraps around according to the periodic setting ATPER. The timer period value must be set before setting this bit. Not all devices contain the event signal output. See the chip configuration details.
3 OFFRST	Reset Timer On Offset Event 0b - The timer is not affected and no action occurs, besides clearing OFFEN, when the offset is reached. 1b - If OFFEN is set, the timer resets to zero when the offset setting is reached. The offset event does not cause a timer interrupt.
2 OFFEN	Enable One-Shot Offset Event 0b - Disable. 1b - The timer can be reset to zero when the given offset time is reached (offset event). The field is cleared when the offset event is reached, so no further event occurs until the field is set again. The timer offset value must be set before setting this field.
1 —	Reserved
0 EN	Enable Timer 0b - The timer stops at the current value. 1b - The timer starts incrementing.

41.5.1.88 Timer Value Register (ATVR)

41.5.1.88.1 Offset

Register	Offset
ATVR	404h

41.5.1.88.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ATIME															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ATIME															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

41.5.1.88.3 Fields

Field	Function
31-0 ATIME	A write sets the timer. A read returns the last captured value. To read the current value, issue a capture command (i.e., set ATCR[CAPTURE]) prior to reading this register.

41.5.1.89 Timer Offset Register (ATOFF)

41.5.1.89.1 Offset

Register	Offset
ATOFF	408h

41.5.1.89.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	OFFSET															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	OFFSET															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

41.5.1.89.3 Fields

Field	Function
31-0 OFFSET	Offset value for one-shot event generation. When the timer reaches the value, an event can be generated to reset the counter. If the increment value in ATINC is given in true nanoseconds, this value is also given in true nanoseconds.

41.5.1.90 Timer Period Register (ATPER)

41.5.1.90.1 Offset

Register	Offset
ATPER	40Ch

41.5.1.90.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PERIOD															
Reset	0	0	1	1	1	0	1	1	1	0	0	1	1	0	1	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PERIOD															
Reset	1	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0

41.5.1.90.3 Fields

Field	Function
31-0 PERIOD	Value for generating periodic events Value for generating periodic events. Each instance the timer reaches this value, the period event occurs and the timer restarts. If the increment value in ATINC is given in true nanoseconds, this value is also given in true nanoseconds. The value should be initialized to 1,000,000,000 (1×10^9) to represent a timer wrap around of one second. The increment value set in ATINC should be set to the true nanoseconds of the period of clock ts_clk, hence implementing a true 1 second counter. **NOTE:** The value of PERIOD has the following constraint: 2^{32} – ENET_ATINC[INC_COR] – 3×ENET_ATINC[INC] ≥ PERIOD > 0.

41.5.1.91 Timer Correction Register (ATCOR)

41.5.1.91.1 Offset

Register	Offset
ATCOR	410h

41.5.1.91.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	COR														
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

41.5.1.91.3 Fields

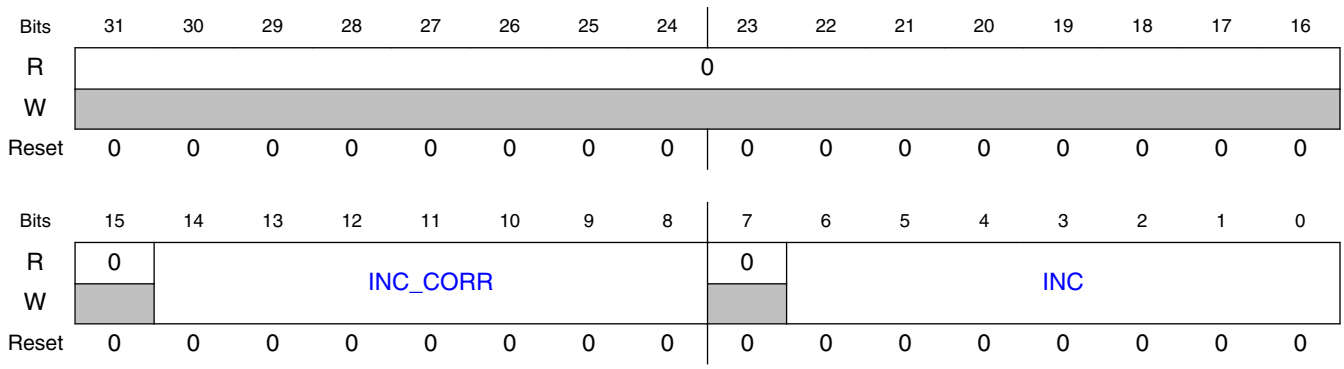
Field	Function
31 —	Reserved
30-0 COR	Correction Counter Wrap-Around Value Defines after how many timer clock cycles (ts_clk) the correction counter should be reset and trigger a correction increment on the timer. The amount of correction is defined in ATINC[INC_CORR]. A value of 0 disables the correction counter and no corrections occur. **NOTE:** This value is given in clock cycles, not in nanoseconds as all other values.

41.5.1.92 Time-Stamping Clock Period Register (ATINC)

41.5.1.92.1 Offset

Register	Offset
ATINC	414h

41.5.1.92.2 Diagram



41.5.1.92.3 Fields

Field	Function
31-15 —	Reserved
14-8 INC_CORR	Correction Increment Value This value is added every time the correction timer expires (every clock cycle given in ATCOR). A value less than INC slows down the timer. A value greater than INC speeds up the timer.
7 —	Reserved
6-0 INC	Clock Period Of The Timestamping Clock (ts_clk) In Nanoseconds The timer increments by this amount each clock cycle. For example, set to 10 for 100 MHz, 8 for 125 MHz, 5 for 200 MHz. **NOTE:** For highest precision, use a value that is an integer fraction of the period set in ATPER.

41.5.1.93 Timestamp of Last Transmitted Frame (ATSTMP)

41.5.1.93.1 Offset

Register	Offset
ATSTMP	418h

41.5.1.93.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TIMESTAMP															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TIMESTAMP															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

41.5.1.93.3 Fields

Field	Function
31-0 TIMESTAMP	Timestamp of the last frame transmitted by the core that had TxBD[TS] set the ff_tx_ts_frm signal asserted from the user application Timestamp of the last frame transmitted by the core that had TxBD[TS] set . This register is only valid when EIR[TS_AVAIL] is set.

41.5.1.94 Timer Global Status Register (TGSR)

41.5.1.94.1 Offset

Register	Offset
TGSR	604h

41.5.1.94.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												TF3	TF2	TF1	TF0
W													W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

41.5.1.94.3 Fields

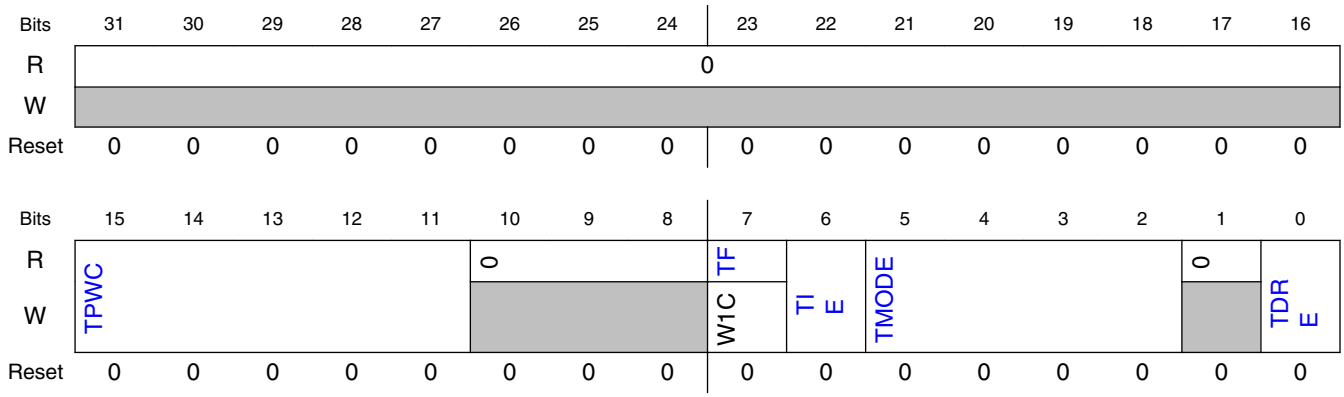
Field	Function
31-4 —	Reserved
3 TF3	Copy Of Timer Flag For Channel 3 0b - Timer Flag for Channel 3 is clear 1b - Timer Flag for Channel 3 is set
2 TF2	Copy Of Timer Flag For Channel 2 0b - Timer Flag for Channel 2 is clear 1b - Timer Flag for Channel 2 is set
1 TF1	Copy Of Timer Flag For Channel 1 0b - Timer Flag for Channel 1 is clear 1b - Timer Flag for Channel 1 is set
0 TF0	Copy Of Timer Flag For Channel 0 0b - Timer Flag for Channel 0 is clear 1b - Timer Flag for Channel 0 is set

41.5.1.95 Timer Control Status Register (TCSR0 - TCSR3)

41.5.1.95.1 Offset

Register	Offset
TCSR0	608h
TCSR1	610h
TCSR2	618h
TCSR3	620h

41.5.1.95.2 Diagram



41.5.1.95.3 Fields

Field	Function
31-16 —	Reserved
15-11 TPWC	Timer PulseWidth Control Specifies the pulse width associated with TMODE values of 1110 or 11X1. Updating this field takes a few cycles to register because it is synchronized to the 1588 clock. When changing this field: 1. Always disable the channel and read the TMODE field to verify that the channel is disabled. 2. Set TPWC to the desired value. 3. Reenable the channel. 00000b - Pulse width is one 1588-clock cycle. 00001b - Pulse width is two 1588-clock cycles. 00010b - Pulse width is three 1588-clock cycles. 00011b - Pulse width is four 1588-clock cycles. 11111b - Pulse width is 32 1588-clock cycles.
10-8 —	Reserved
7 TF	Timer Flag Sets when input capture or output compare occurs. This flag is double buffered between the module clock and 1588 clock domains. When this field is 1, it can be cleared to 0 by writing 1 to it. 0b - Input Capture or Output Compare has not occurred. 1b - Input Capture or Output Compare has occurred.
6 TIE	Timer Interrupt Enable 0b - Interrupt is disabled 1b - Interrupt is enabled
5-2 TMODE	Timer Mode Updating the Timer Mode field takes a few cycles to register because it is synchronized to the 1588 clock. The version of Timer Mode returned on a read is from the 1588 clock domain. When changing Timer Mode, always disable the channel and read this register to verify the channel is disabled first. 0000b - Timer Channel is disabled. 0001b - Timer Channel is configured for Input Capture on rising edge.

Table continues on the next page...

Field	Function
	0010b - Timer Channel is configured for Input Capture on falling edge. 0011b - Timer Channel is configured for Input Capture on both edges. 0100b - Timer Channel is configured for Output Compare - software only. 0101b - Timer Channel is configured for Output Compare - toggle output on compare. 0110b - Timer Channel is configured for Output Compare - clear output on compare. 0111b - Timer Channel is configured for Output Compare - set output on compare. 1000b - Reserved 1010b - Timer Channel is configured for Output Compare - clear output on compare, set output on overflow. 10x1b - Timer Channel is configured for Output Compare - set output on compare, clear output on overflow. 110xb - Reserved 1110b - Timer Channel is configured for Output Compare - pulse output low on compare for 1 to 32 1588-clock cycles as specified by TPWC. 1111b - Timer Channel is configured for Output Compare - pulse output high on compare for 1 to 32 1588-clock cycles as specified by TPWC.
1 —	Reserved
0 TDRE	Timer DMA Request Enable 0b - DMA request is disabled 1b - DMA request is enabled

41.5.1.96 Timer Compare Capture Register (TCCR0 - TCCR3)

41.5.1.96.1 Offset

Register	Offset
TCCR0	60Ch
TCCR1	614h
TCCR2	61Ch
TCCR3	624h

41.5.1.96.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TCC															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TCC															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

41.5.1.96.3 Fields

Field	Function
31-0 TCC	Timer Capture Compare This register is double buffered between the module clock and 1588 clock domains. When configured for compare, the 1588 clock domain updates with the value in the module clock domain whenever the Timer Channel is first enabled and on each subsequent compare. Write to this register with the first compare value before enabling the Timer Channel. When the Timer Channel is enabled, write the second compare value either immediately, or at least before the first compare occurs. After each compare, write the next compare value before the previous compare occurs and before clearing the Timer Flag. The compare occurs one 1588 clock cycle after the IEEE 1588 Counter increments past the compare value in the 1588 clock domain. If the compare value is less than the value of the 1588 Counter when the Timer Channel is first enabled, then the compare does not occur until following the next overflow of the 1588 Counter. If the compare value is greater than the IEEE 1588 Counter when the 1588 Counter overflows, or the compare value is less than the value of the IEEE 1588 Counter after the overflow, then the compare occurs one 1588 clock cycle following the overflow. When configured for capture, the value of the IEEE 1588 Counter is captured into the 1588 clock domain and then updated into the module clock domain, provided the Timer Flag is clear. Always read the capture value before clearing the Timer Flag.