Design Gateway Co.,Ltd
E-mail: ip-sales@design-gateway.com
URL: www.design-gateway.com

Core Facts

Provided with Core	
Documentation	User Guide, Design Guide
Design File Formats	Encrypted hdl File
Instantiation Templates	VHDL
Reference Designs & Application Notes	QuartusII Project, See Reference Design Manual
Additional Items	Demo on Arria10 GX development board
Support	
Support Provided by Design Gateway Co., Ltd.	

Features

- 10 Gbps Ethernet MAC and PCS
- Directly connecting with Intel transceiver PHY (PMA) by 32-bit interface
- Low latency solution: 65.1 ns for round-trip latency (18.6 ns for Tx path, 21.7 ns for Rx path, and 24.8 ns for PMA latency)
- Avalon-stream interface with the user logic
- Small resource utilization
- Minimum Tx packet size: 5 bytes
- FCS (CRC-32) inserting and checking
- 64B/66B Encoding and Decoding following IEEE802.3ae specification
- Supporting 10GBASE-R standard
- Appending zero padding for Tx interface, but not removing zero padding for Rx interface
- Individual clock domain for transmit and receive interface at 322.265625 MHz
- Reference design available on Intel development board (Arria10 GX)

Table 1: Example Implementation Statistics

Family	Example Device	Fmax (MHz)	ALMs[1]	Registers[1]	Pin	Block Memory bit[2]	Design Tools
Arria10 GX	10AX115S2F45I2SG	322	1017	1113	-	-	QuartusII 18.0

Notes:

1) Actual logic resource dependent on percentage of unrelated logic

Applications

Nowadays the network with low latency access is the core for many real-time applications such as High Frequency Trading (HFT), Data center, and Real-time control system in Industrial and Automotive. LL10GEMAC is designed to implement Ethernet MAC and PCS for low latency networking by 10Gb Ethernet. The example application of HFT by using LL10GEMAC is shown in Figure 1.

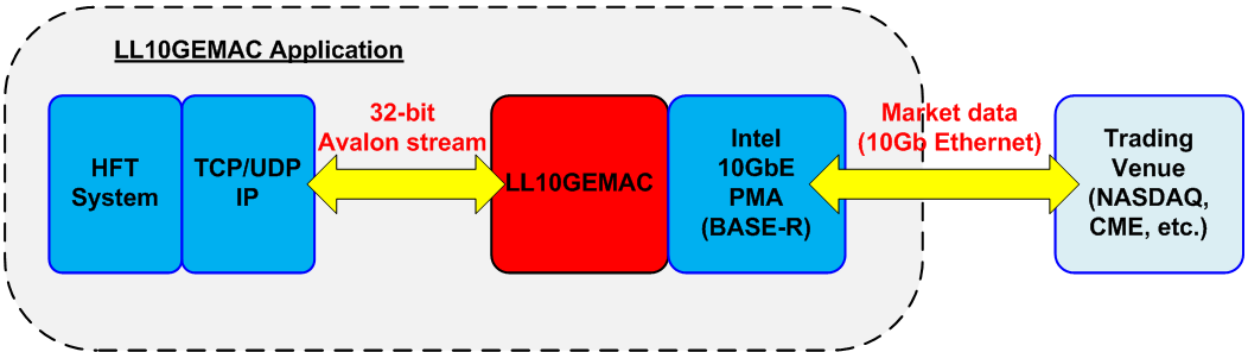

Figure 1: LL10GEMAC Application

Reference design

LL10GEMAC IP has been launched with the loopback demo design on Intel development board. In the demo, Test logic sends small packets to LL10GEMAC IP and then waits until the packet is returned to Test logic via SFP+ loopback. Data latency of the system can be measured in the demo including the latency in LL10GEMAC. As shown in Figure 2, LL10GEMAC IP shows the very low round-trip latency measured from Tx path to Rx path of Avalon-stream interface which is 65.1 ns (21 clock cycles of 322.265625 MHz).

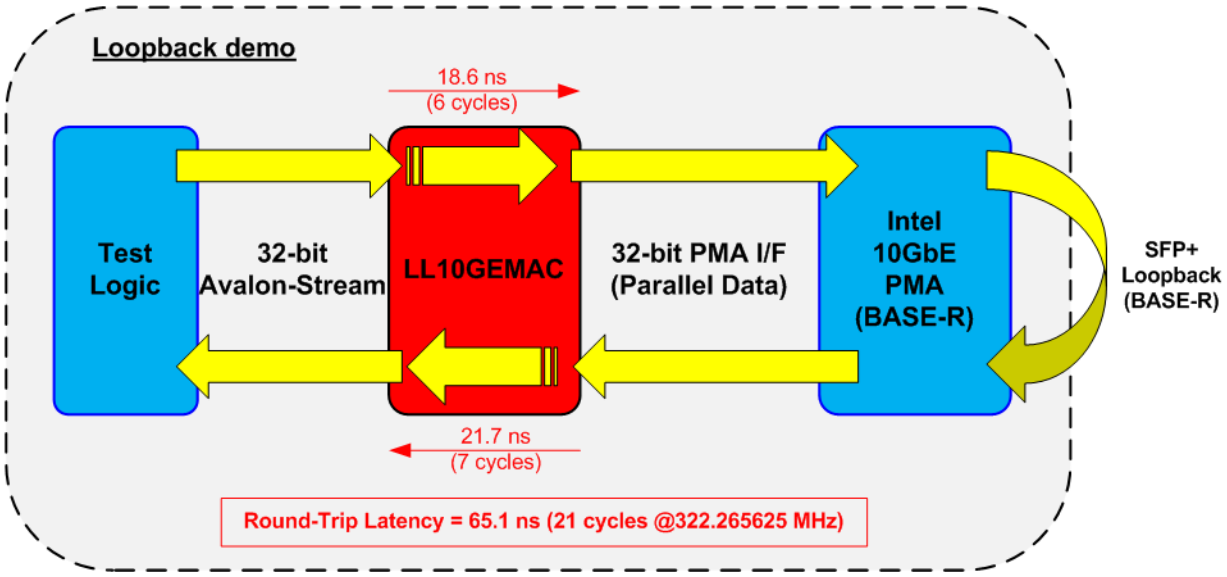

Figure 2: LL10GEMAC latency

General Description

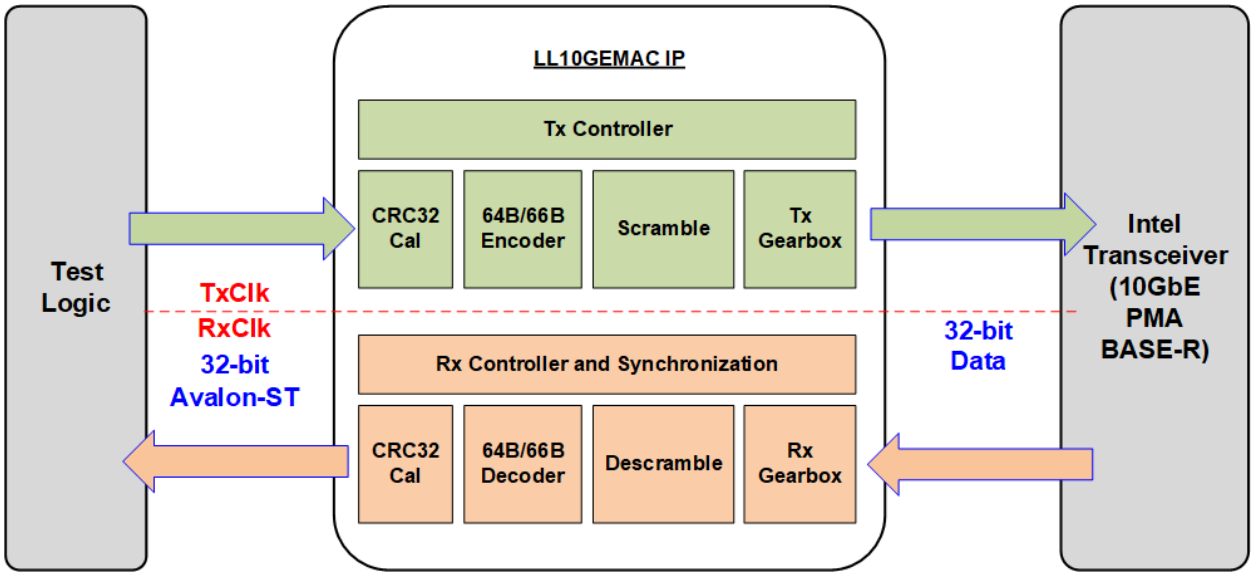

Figure 3: LL10GEMAC Block diagram

LL10GEMAC IP implements the MAC layer and the PCS (Physical Coding Sublayer) layer for 10Gb Ethernet solution. After power up, Rx controller and synchronization is run to calibrate the receive interface with PMA (Intel transceiver) until the received data can be locked. Next, the received data from PMA is monitored to check link up status. After the connection is ready, LL10GEMAC IP asserts ready signal to user interface and then the user can transmit the packet to LL10GEMAC IP.

For transmitting the packet, the preamble and SFD (Start frame delimiter) are appended to be the packet header by LL10GEMAC IP. At the end of packet, the zero padding (when the packet size is very small), FCS (Frame check sequence), and IFG (Interframe gap) are appended to be the packet footer. Next, the packet is encoded by 64B/66B encoder and then scrambled by Scramble block. Finally, the data is forwarded to Tx Gearbox module for transmitting 32-bit data to the PMA.

On the other hand, the received data from the PMA is re-aligned by Rx Gearbox as the first step. After that, the data is decrambled and decoded by Descramble and 64B/66B decoder block respectively. Finally, FCS of the packet is verified and removed with the preamble and SFD. Only Ethernet data is forwarded to Avalon-stream interface (Avalon-ST I/F). If the FCS is incorrect, the IP asserts the error signal to Avalon-ST I/F.

When the user sends one packet to LL10GEMAC IP, the data of each packet must be always ready until end of packet. Data valid must be always asserted to ‘1’ during the packet transmission. However, ready signal from LL10GEMAC IP can be de-asserted to ‘0’ to pause data transmission for 1 clock cycle every 32 clock cycles to match with 64B/66B encoder characteristic.

In the same way, the user must be always ready to receive the packet from LL10GEMACIP because there is no receive buffer inside LL10GEMAC IP. During a packet transmission, the data valid of the received packet is de-asserted to ‘0’ for 1 clock cycle every 32 clock cycles to pause data transmission to match with 64B/66B decoder characteristic.

Functional Description

As shown in Figure 3, LL10GEMAC IP supports data transmission in both directions at the same time. Tx and Rx logics are run independently in the different clock domain.

Transmit Block

The data packet from Avalon-ST is fed to LL10GEMAC IP for CRC32 calculation (FCS), 64B/66B Encoding, Scrambling, and Aligning before forwarding to the PMA.

- Tx Controller

Tx Controller inserts the packet header and the packet footer to the packet. The header consists of 7-byte preamble and 1-byte SFD (Start of frame delimiter) while the footer consists of 4-byte FCS (CRC-32) and 1-byte EFD (End of frame delimiter). If the packet is too short, zero-padding is also appended to the packet after the last Ethernet data. After finishing one frame transmission, Idle is inserted to be IFG (Interframe Gap) of each packet.

Furthermore, Tx controller monitors the control signals on Avalon-ST to detect a new frame transmitting from the user. Also, the ready signal on Avalon-ST is de-asserted to insert the header for 64B/66B encoding with Gearbox operating.

- CRC32 Cal

This module is designed to calculate 32-bit CRC of one data packet by using 32-bit data bus, input from Avalon-stream interface. The polynomial of CRC-32 to create FCS following IEEE802.3ae standard is shown as follows.

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

FCS is appended to be the packet footer after finishing transferring the Ethernet data that may include zero-padding.

- 64B/66B Encoder

64B/66B data encoding is applied in 10Gb Ethernet transmission for tracking the clock in the clock data recovery (CDR) module. 64B/66B encoding logic is designed to have less overhead and support only 10GBASE-R following IEEE802.3ae specification. The packet with appending the header and the footer is encoded by this module before forwarding to Scramble module.

- Scramble

Encoded data must be scrambled to avoid long sequences of ‘1’s and ‘0’s. According to IEEE802.3ae standard, the polynomial of scrambling is as follows.

$$P(X) = X^{58} + X^{39} + 1$$

- Tx Gearbox

Before sending data to PMA, the data output from 64B/66B encoder and scramble data is re-aligned to 32-bit parallel data. To match the bandwidth, the data from the user is paused for 1 clock cycle every 32 clock cycles for inserting the header inside 64B/66B encoder.

Receive Block

The submodule inside Receive Block has the reversed operation from Transmit Block. The additional feature of Receive Block is Synchronization block.

- Rx Gearbox

Rx Gearbox receives the 32-bit parallel data from the PMA and then separates into 2-bit header and 64-bit data without considering data alignment. This block includes bit-slip logic to re-align the data until the received data bus is in correct order.

- Rx Controller and Synchronization

After finishing the reset sequence, Synchronization monitors the received data from Rx Gearbox for tuning data alignment by sending slip signal until the data can be locked correctly. After the Ethernet connection is linked up, the data alignment is still monitored for re-tuning if the unalignment is found.

After the received packet from Rx Gearbox is descrambled and decoded, SFD and FCS of the packet are verified. The controller generates the error to Avalon-ST when some errors are found in the received packet. The header and the footer are removed from the packet before forwarding to Avalon-ST. However, zero-padding is not removed from the packet to minimize the latency time in Rx path.

- Descramble

The process when receiving the data from Rx Gearbox is de-scrambled. Next, the descrambled data is forwarded to 64B/66B Decoder module.

- 64B/66B Decoder

This module decodes the descrambled data to check link up status, start of frame, and end of frame. The outputs of the module are the data and the data type which are monitored by Rx controller to validate the data sequence in the packet.

- CRC32 Cal

This module is the same module as CRC32 Cal in Transmit Block, but the calculated CRC32 is applied to verify the FCS extracted from the received packet. The error is asserted on Rx Avalon-ST if the FCS in the received packet is not correct.

10GbE PMA (BASE-R)

This module is created by using Native PHY IP to configure Intel transceiver to run as PMA module of 10Gb Ethernet BASE-R standard. The IP is configured to be PCS direct mode and has 32-bit parallel data running at 322.265625 MHz. For more details, please download the IP datasheet from the following link.

https://www.intel.com/content/www/us/en/programmable/products/intellectual-property/ip/interface-protocols/m-alt-10gbase-r-pcs.html

Core I/O Signals

Descriptions of all signal I/Os are provided in Table 2.

Table 2: Core I/O Signals

Signal	Dir	Description
User Interface		
Linkup	Out	'1'-Link up, '0'-Link down. Assert to '1' when Ethernet connection is established and PMA returns Idle code in receive interface. This signal is synchronous to RxClk.
TxTestPin[7:0]	Out	Reserved to be IP Test point. Synchronous to TxClk.
RxTestPin[7:0]	Out	Reserved to be IP Test point. Synchronous to RxClk.
IPVersion[31:0]	Out	IP version number
Tx Avalon Stream interface (Synchronous to TxClk)		
MacTxData[31:0]	In	Transmitted data to Avalon-ST interface. Valid when MacTxValid is asserted to '1'.
MacTxEmpty[1:0]	In	Specify the number of bytes which are unused of the final MacTxData in the frame. The signal is valid when MacTxValid and MacTxEOP are asserted to '1'.
MacTxValid	In	Transmit data valid signal. Assert to '1' when MacTxData is valid.
MacTxSOP	In	Assert to '1' to indicate the first word in the frame. Valid when MacTxValid is asserted to '1'.
MacTxEOP	In	Assert to '1' to indicate the final word in the frame. Valid when MacTxValid is asserted to '1'.
MacTxReady	Out	Handshaking signal. Asserted to '1' when MacTxData has been accepted. When the signal is de-asserted to '0' to pause data transmission, MacTxData/MacTxEmpty/MacTxValid/MacTxEOP must be latched to the same value until MacTxReady is re-asserted to '1'.
Rx Avalon stream interface (Synchronous to RxClk)		
MacRxData[31:0]	Out	Received data. Valid when MacRxValid is asserted to '1'.
MacRxEmpty[1:0]	Out	Specify the number of bytes which are unused of the final MacRxData in the frame. The signal is valid when MacRxValid and MacRxEOP are asserted to '1'.
MacRxValid	Out	Asserted to '1' when the received data is valid. During packet transmission, this signal may be de-asserted to '0' to pause data transmission.
MacRxSOP	Out	Assert to '1' to indicate the first word in the frame. Valid when MacRxValid is asserted to '1'.
MacRxEOP	Out	Assert to '1' to indicate the final word in the frame. Valid when MacRxValid is asserted to '1'.
MacRxError	Out	Asserted at the end of the frame (MacRxEOP='1' and MacRxValid='1') to indicate that the frame has an error. '0': normal packet, '1': error packet. The packet is error when SFD, FCS, or EFD is not correct.
Tx PMA I/F (Synchronous to TxClk)		
TxRstB	In	Reset IP core in TxClk domain.
TxClk	In	Clock output from the PMA for Tx interface. 322.265625 MHz for 32-bit interface.
PMATxData[31:0]	Out	32-bit transmitted data to the PMA.
Rx PMA I/F (Synchronous to RxClk)		
RxRstB	In	Reset IP core in RxClk domain.
RxClk	In	Clock output from the PMA for Rx interface. 322.265625 MHz for 32-bit interface.
PMARxData[31:0]	In	32-bit data received from the PMA.

Timing Diagram

IP Initialization

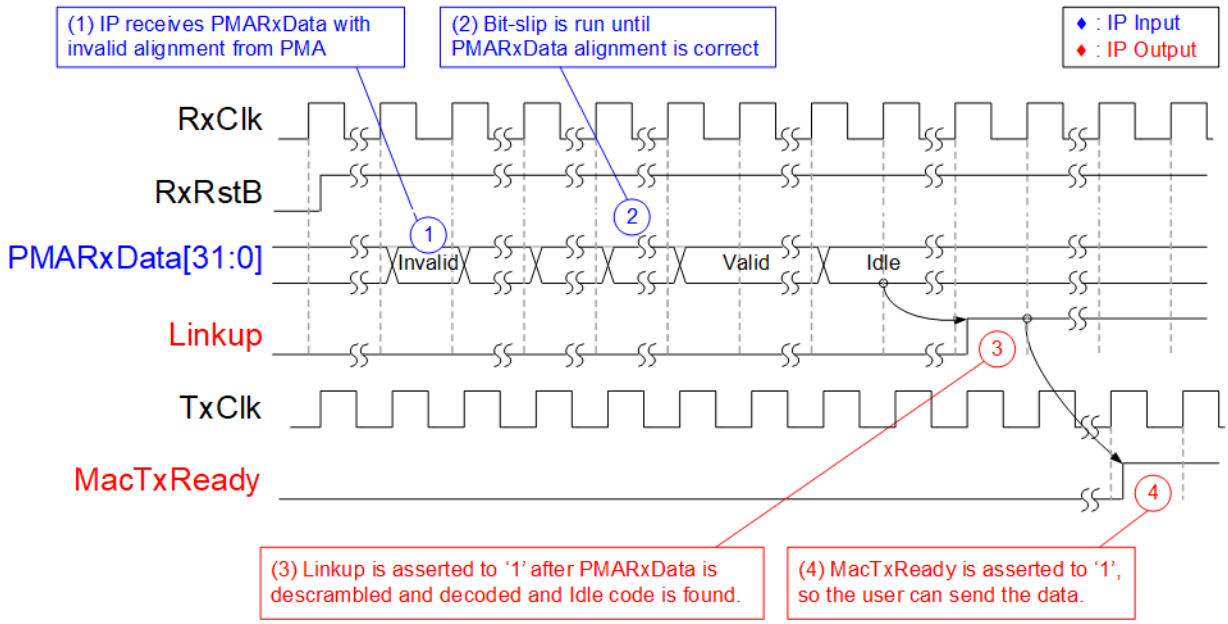

Figure 4: Rx Tuning and Linkup timing diagram

After RxRstB is asserted to '1', the receive module inside the IP begins synchronization process to tune and lock the received data from the PMA. After the data is locked and the ethernet connection can be established, Linkup is asserted to '1' in RxClk domain. Finally, MacTxReady is asserted to '1' in TxClk domain. More details of the initialization process are described as follows.

(1) IP receives PMARxData from the PMA every clock cycle, but the data alignment is still not corect.
(2) The received data is tuned until the data is valid format. The synchronization process is now completed.
(3) The received data is descrambled and decoded. Linkup signal is asserted to '1' when the Idle code is detected by IP.
(4) The IP is ready to transfer the data from user by asserting MacTxReady to '1'.

Transmit interface

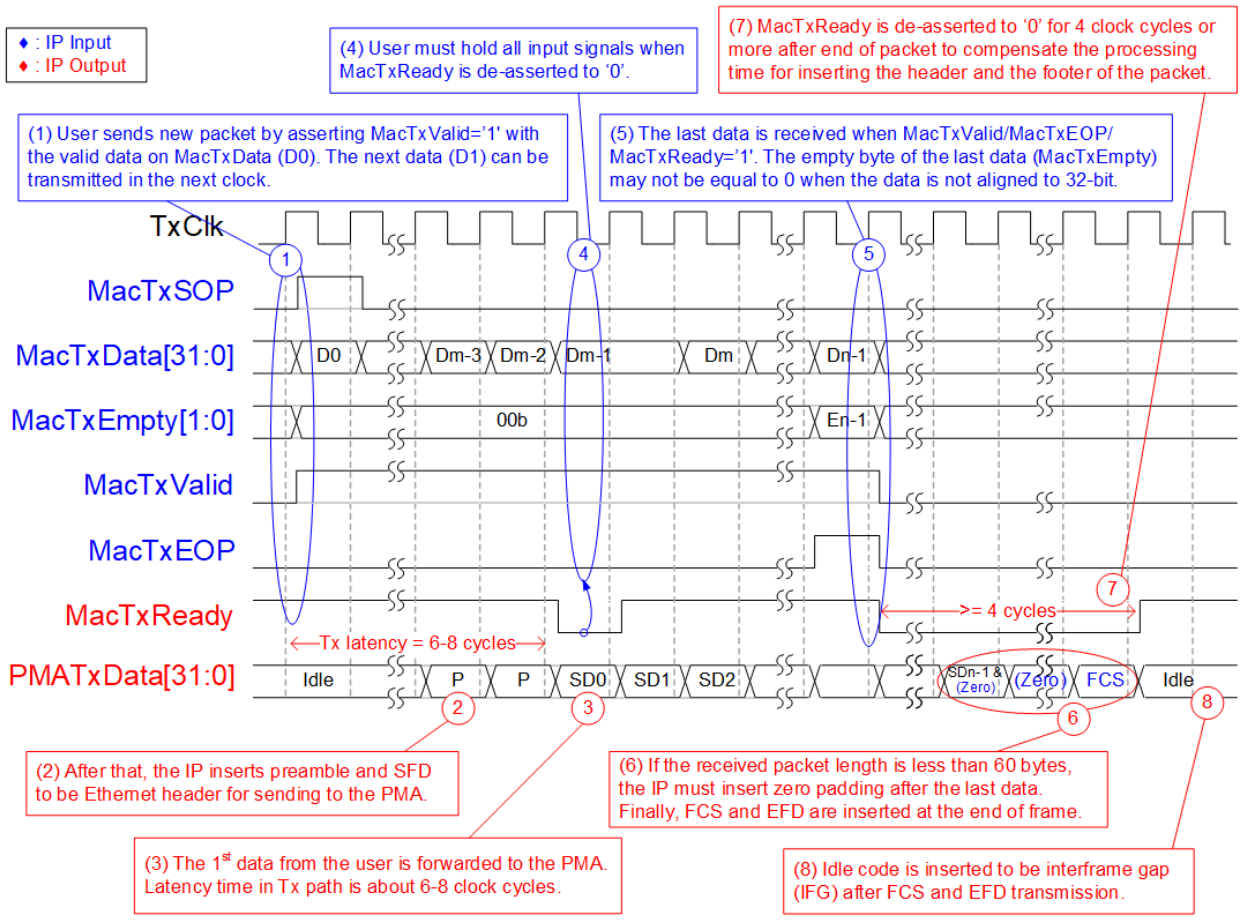

Figure 5: Transmit interface timing diagram

When the new frame is transmitted from the user, the IP inserts the header of the packet - Preamble and SFD before transmitting the data. Also, the FCS and EFD are inserted as the packet footer by the IP. All data including the header and the footer, output from the IP, are encoded, scrambled, and re-aligned to 32-bit parallel data. If the packet length is too short (less than 60-byte), zero-padding must be inserted before FCS transmission.

Note: Idle code, Preamble, Data, and FCS on 32-bit PMATxData cannot be read without using the receiver module to align, de-scramble, and decode the data to be the raw value.

(1) The new packet is found when MacTxValid is asserted to '1' with the IP ready status (MacTxReady='1'). MacTxEmpty must be always equal to 00b except the last data which may not equal to 00b for 32-bit unaligned data.
Note: Sometimes, MacTxReady can be de-asserted to '0' about 1-2 clock cycles after receiving the first data. So, the second data (D1) must be holded the value until MacTxReady is re-asserted to '1'.
(2) After that, 7-byte preamble and SFD are sent to the PMA.
(3) The 1st data is sent by the IP to the PMA. Data latency of Tx path, measured by the first data on MacTxData and the first data on PMATxData, is about 6-8 clock cycles, depending on the data sequence in the Gearbox.
(4) When MacTxReady is de-asserted to '0', all input signals from user (MacTxData, MacTxEmpty, MacTxValid, MacTxSOP, and MacTxEOP) must hold the same value until MacTxReady is re-asserted to '1'. Typically, MacTxReady is de-asserted to '0' for one cycle every 32 clock cycles to pause data transmission to add the header following 64B/66B encoding process.
(5) After the last data is found (MacTxValid/MacTxEOP/MacTxReady='1'), the IP ready (MacTxReady) is de-asserted to '0' to pause data transmission for waiting the IP to finish the packet post-processing. In this cycle, MacTxEmpty shows the unused byte of the last data which may be equal to 00b (4-byte valid), 01b (3-byte valid), 10b (2-byte valid), or 11b (1-byte valid).
(6) The last data which is encoded and encrypted by the IP is transmitted to the PMA. Zero-padding is inserted when the packet is too short.
(7) After the last data is received from the user, MacTxReady is de-asserted at least 4 clock cycles. The number of output data from the IP is more than the number of input data from the user due to the appending packet header and footer. So, MacTxReady is de-asserted to compensate the different value between the number of input data and output data.
(8) Idle code is always inserted at the end of packet to be the interframe gap (IFG) of the packet. At least 9-byte IFG is inserted by the IP.

Receive Interface

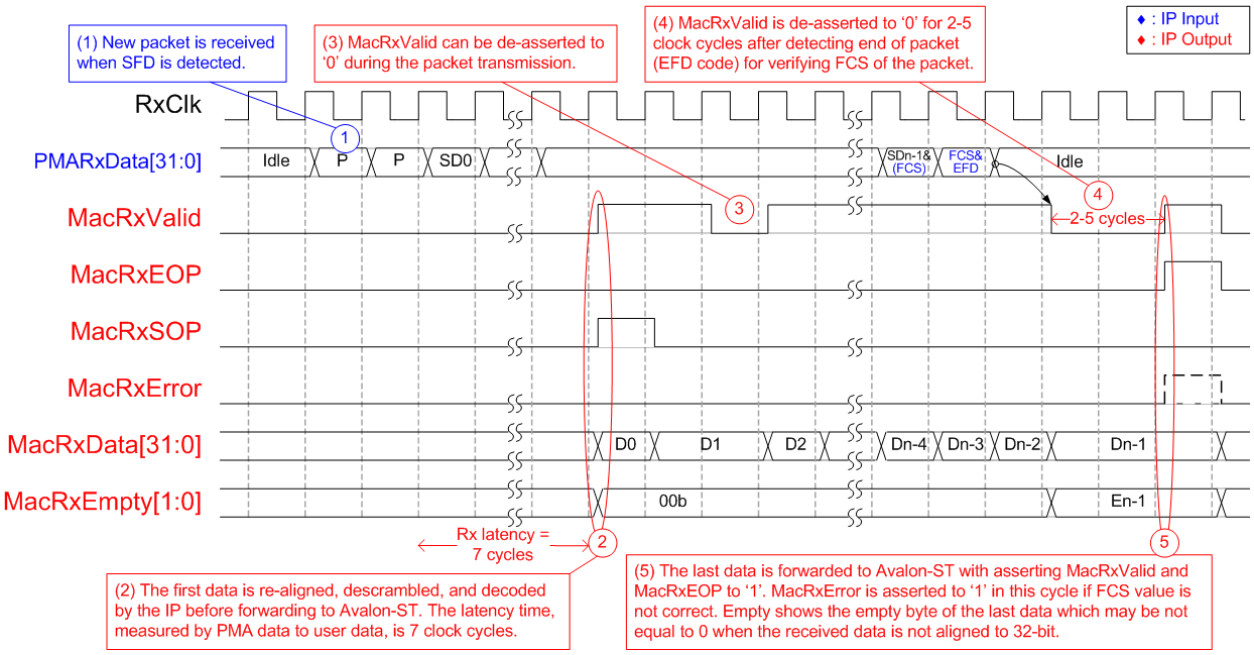

Figure 6: Receive interface timing diagram

When the new packet is received from the PMA, the IP re-aligns 32-bit data and then descrambles and decodes to check the start of packet and the end of the packet. The header and the footer of the packet are verified and removed before forwarding to the user. If SFD, FCS, or EFD code is not correct, error signal (MacRxError='1') is asserted to the user.

(1) The IP begins the operation after detecting SFD code on the decoded data from Rx PMA interface.
(2) After detecting the first data from the PMA for 7 clock cycles, the IP sends the first data to the user. The latency time of the data bus is applied for converting data stream to be raw data for the user. The data on the user interface is valid for all 32 bits except the last data which may be valid only some bytes. So, MacRxEmpty is always equal to 0, except the last data. MacRxEmpty of the last data may be equal to 00b (4-byte valid), 01b (3-byte valid), 10b (2-byte valid), or 11b (1-byte valid).
(3) The valid signal of user interface (MacRxValid) is de-asserted to '0' for 1 clock cyle every 32 clock cycles to remove the header from the data following 64B/66B decoding process.
(4) After the IP detects end of packet (EFD code), MacRxValid is de-asserted to '0' for 2-5 clock cycles for verifying FCS of the received packet.
(5) The IP asserts MacRxEOP and MacRxValid to '1' with the last data on MacRxData. At the same time, MacRxError is de-asserted to '0' when the error is not detected in the received packet. Otherwise, MacRxError is asserted to '1'.

The IP removes the packet header and footer from the PMA before forwarding to the user, but zero-padding is not removed to optimize latency time.

Verification Methods

The LL10GEMAC IP Core functionality was verified by simulation and also proved on real board design by using Arria10 GX development board.

Recommended Design Experience

User must be familiar with HDL design methodology to integrate this IP into system.

Ordering Information

This product is available directly from Design Gateway Co., Ltd. Please contact Design Gateway Co., Ltd. For pricing and additional information about this product using the contact information on the front page of this datasheet.

Revision History

Revision	Date	Description
1.0	26-Mar-2021	New release