Design Gateway Co.,Ltd

E-mail: ip-sales@design-gateway.com
URL: www.design-gateway.com

Core Facts

Provided with Core	
Documentation	Reference Design Manual Demo Instruction Manual
Design File Formats	Encrypted netlist file
Instantiation Templates	VHDL
Reference Designs & Application Notes	Vivado Project, See Reference Design Manual
Additional Items	Demo on ZCU102
Support	
Support Provided by Design Gateway Co., Ltd.	

Features

- 10 Gbps Ethernet MAC and PCS
- Directly connecting with 32-bit PMA by Xilinx IP wizard.
- 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)
- AXI4-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 Xilinx development board (ZCU102)

Table 1: Example Implementation Statistics

Family	Example Device	Fmax (MHz)	CLB Regs	CLB LUTs	CLB	IOB	BRAMTile	Design Tools
Kintex-Ultrascale	XCKU040FFVA1156-2E	322.266	1093	1366	263	-	-	Vivado2019.1
Zynq-Ultrascale+	XCZU7EV-FFVC1156-2E	322.266	1093	1361	269	-	-	Vivado2019.1
Virtex-Ultrascale+	XCVU9P-FLGA2104-2L	322.266	1093	1362	263	-	-	Vivado2019.1

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.

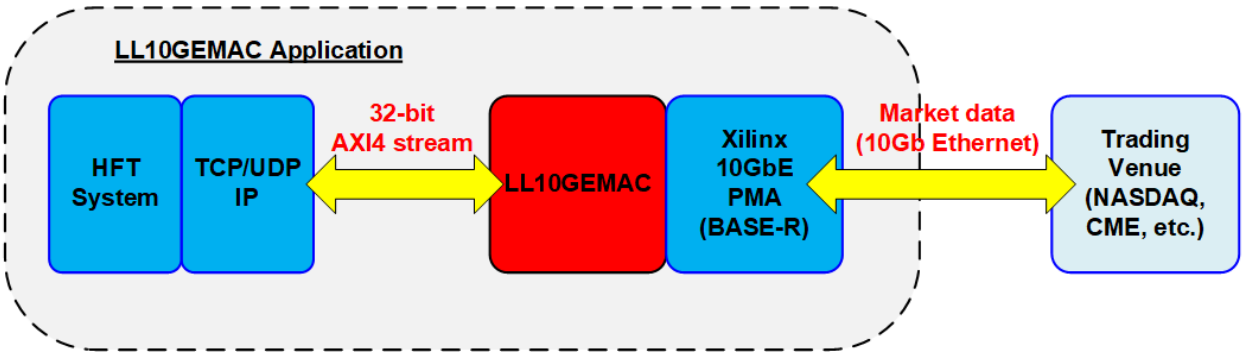

Figure 1: LLGEMAC Application

Reference design

LL10GEMAC IP has been launched with the loopback demo design on Xilinx 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 each module 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 ot Rx path of AXI4-Stream interface, 65.1 ns (21 clock cycles of 322.265625 MHz).

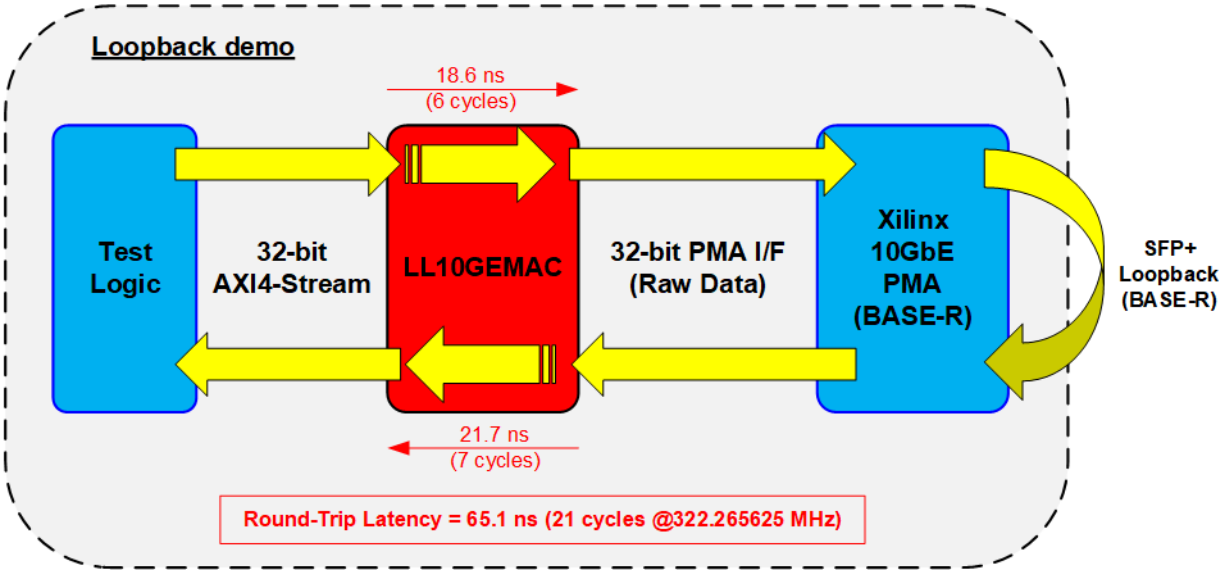

Figure 2: LL10GEMAC latency

According to AR#68177 (https://www.xilinx.com/support/answers/68177.html), TX and Rx Latency Values of UltraScale+ GTH Transceiver depend on the submodule using in the transceiver block. Figure 3 shows the label of the submodule using in our loopback demo for very low-latency solution.

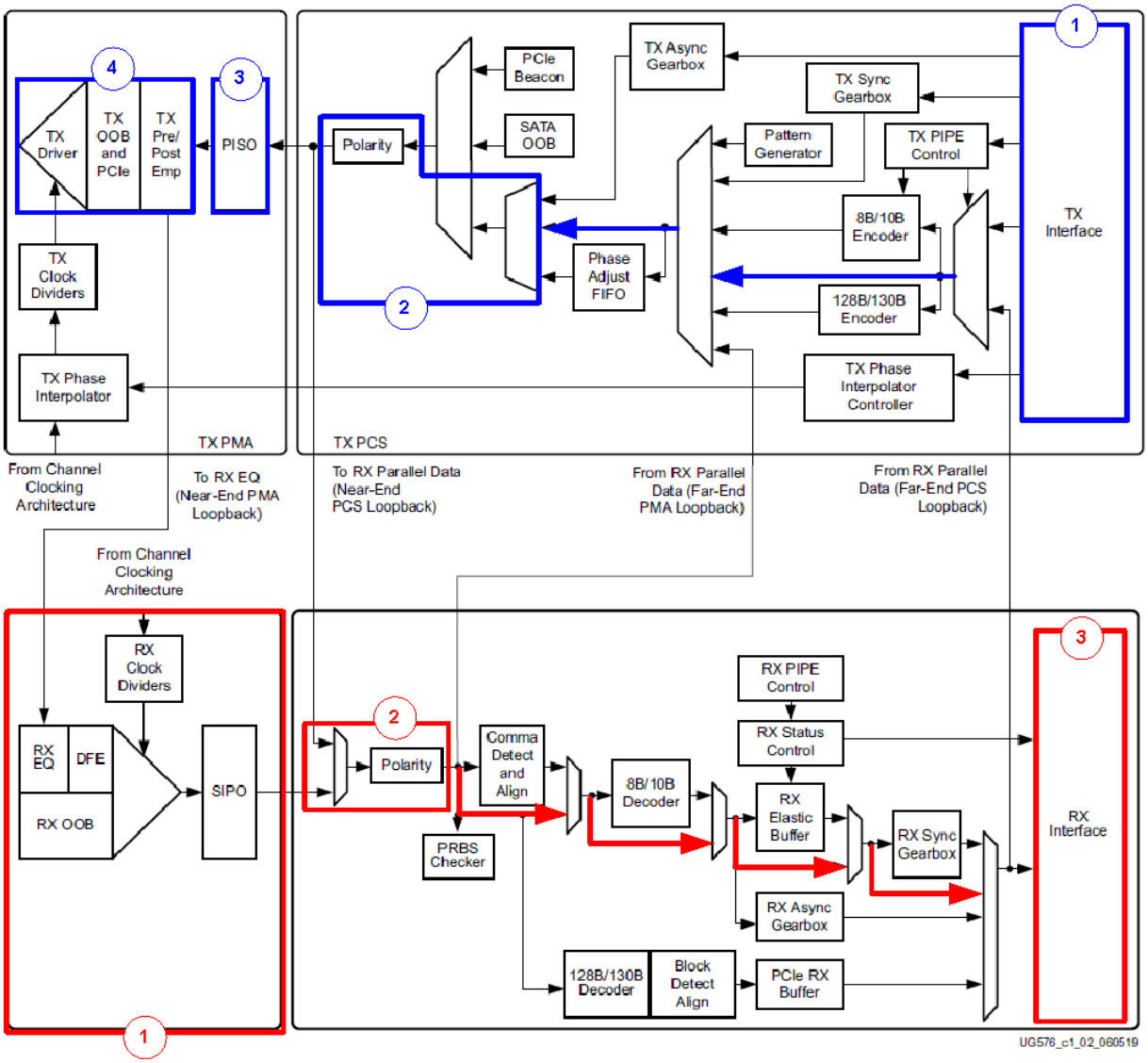

Figure 3: Xilinx PMA for 10GbE by Raw Data (Ref-UG576 UltraScale GTH transceiver)

In conclusion, the latency value of PMA when running the loopback demo are described in more details as follows.

Tx Latency

(1) Tx Fabric Interface = 32 UI
(2) To TX PCS/PMA boundary = 32 UI
(3) To Serializer = 64 UI
(4) PMA = 15 UI

Maximum value of total Tx Latency is 143 UI.

Rx Latency

(1) PMA = 60.5 UI
(2) PMA to PCS = 16 UI
(3) Rx Fabric Interface = 32 UI

Maximum value of total Rx Latency is108.5 UI.

So, maximum latency time of PMA when running loopback demo is 143 + 108.5 UI. In our loopback demo by using 322.265625 MHz clock for measuring latency time on ZCU102 (UltraScale+ GTH transceiver), the total latency time in PMA is about 8 clock cycles or 24.8 ns.

Note: 10 Gb Ethernet transfer speed is 10.3125 Gbps, so 1 UI is equal to 1/10.3125G = 0.097 ns.

General Description

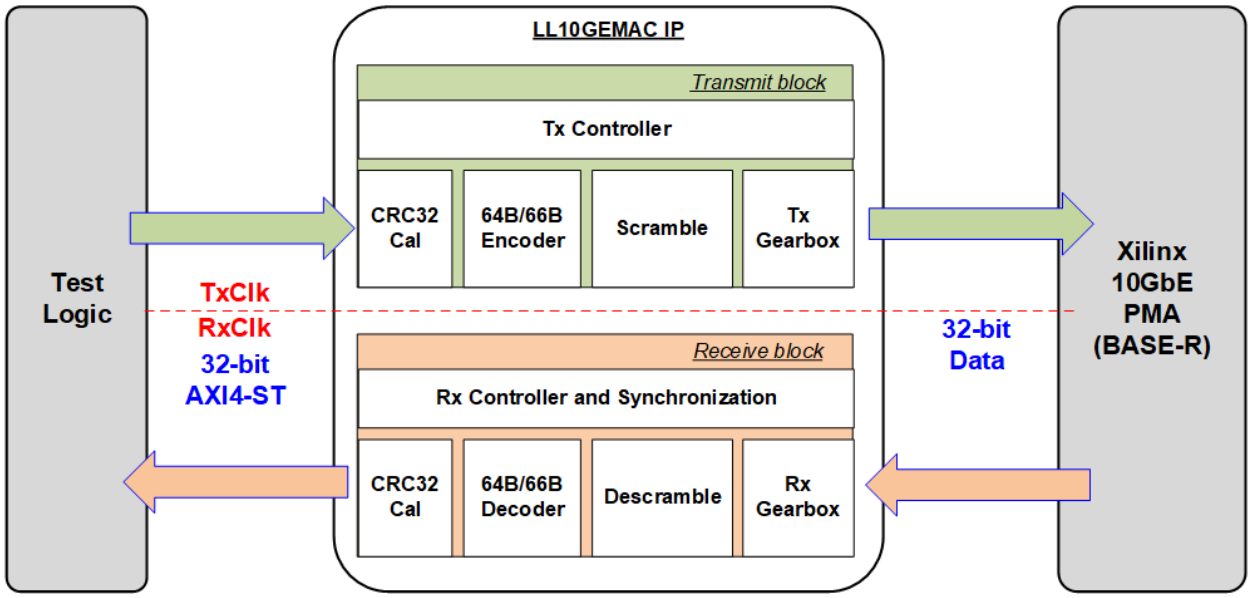

Figure 4: LL10GEMAC Block diagram

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 PMA.

On the other hand, the received packet from the PMA is re-aligned by Rx Gearbox as the first step. After that, the data is descrambled 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 AXI4 stream interface (AXI4-ST I/F). If the FCS is not correct, the IP asserts the error signal to AXI4-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.

Similarly, 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 characterisric.

Functional Description

As shown in Figure 4, 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 AXI4-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 paket. 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 AXI4-ST to detect a new frame transmitting from the user. Also, the ready signal on AXI4-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 AXI4-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

According to 64B/66B operation, every 64-bit data input from user is encoded to 66-bit data. The data output size to PMA is 32-bit. Therefore, Tx Gearbox is the module to re-align the encoded and scrambled data to 32-bit. Since the bandwidth of Scramble module is higher than PMA interface, the data from user must be paused for 1 clock cycle every 32 clock cycles. During pause time, the header of 64B/66B encoder can be inserted.

Receive Block

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

- Rx Gearbox

The 32-bit parallel data stream from the PMA is split to 2-bit header and 64-bit data without considering data alignment. After that, bit-slip logic is operated to re-align the data until the received data bus is in correct order. The control logic of bit-slip function is designed in the next module.

- 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 AXI4-ST when some errors are found in the received packet. The header and the footer are removed from the packet before forwarding to AXI4-ST. However, zero-padding is not removed from the packet to minimize the latency time in Rx path.

- Descramble

This module decrambles the data output from Rx Gearbox before forwarding to 64B/66B Decoder.

- 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 AXI4-ST if the FCS in the received packet is not correct.

10GbE PMA (10GBASE-R)

10GBASE-R PMA, provided by Xilinx without the charge, is generated by using UltraScale FPGAs Transceivers Wizard. The wizard has the template which helps the user to set the transceiver parameters for 10GBASE-R operation. To run with LL10GEMAC, please change the following setting from the default value of BASE-R template.

- Encoding/Decoding : Raw
- Transmitter/Receiver Buffer : Bypass

Please see more details about the Wizard from the following link.

https://www.xilinx.com/products/intellectual-property/ultrascale_transceivers_wizard.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 AXI4 stream interface (Synchronous to TxClk)		
tx_axis_tdata[31:0]	In	Transmitted data of AXI4-stream interface. Valid when tx_axis_tvalid='1'.
tx_axis_tkeep[3:0]	In	Byte enable of 32-bit tx_axi_tdata. Asserted to '1' for one bit when each byte is valid. Bit[0], [1], [2], and [3] are asserted to '1' when tdata[7:0], [15:8], [23:16], and [31:24] are valid respectively. When tx_axis_tvalid='1', tx_axis_tkeep is equal to Fh for sending 32-bit data in each packet except the last data (tx_axis_tlast='1'). The byte enable of the last data can be equal to 1h, 3h, 7h, and Fh when 1 – 4 byte data is valid sequentially.
tx_axis_tvalid	In	Assert to '1' to transmit data. This signal must be always asserted to '1' from start of packet to end of packet. *Note: The minimum size of transmit data from user is 5-bytes (tvalid is asserted to '1' to transfer two data).*
tx_axis_tlast	In	Assert to '1' to indicate the final word in the frame. Valid when tx_axis_tvalid='1'.
tx_axis_tready	Out	Handshaking signal. Asserted to '1' when tx_axis_tdata has been accepted. When the signal is de-asserted to '0' to pause data transmission, tx_axis_tdata/tkeep/tvalid/tlast must be latched to the same value until tx_axis_tready is re-asserted to '1'.
Rx AXI4 stream interface (Synchronous to RxClk)		
rx_axis_tdata[31:0]	Out	Received data. Valid when rx_axis_tvalid='1'.
rx_axis_tkeep[3:0]	Out	Received data byte enable. Asserted to '1' for one bit when each byte is valid. Bit[0], [1], [2], and [3] are asserted to '1' when tdata[7:0], [15:8], [23:16], and [31:24] are valid respectively. The signal is valid when rx_axis_tvalid='1'.
rx_axis_tvalid	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.
rx_axis_tlast	Out	Assert to '1' to indicate the final word in the frame. Valid when rx_axis_tvalid='1'.
rx_axis_tuser	Out	Valid at the end of the frame transmission (rx_axis_tlast='1' and rx_axis_tvalid='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, output from the PMA. Active Low.
TxClk	In	Clock output from the PMA for Tx interface. 322.265625 MHz for 32-bit interface.
TxUserData[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, output from the PMA. Active Low.
RxClk	In	Clock output from the PMA for Rx interface. 322.265625 MHz for 32-bit interface.
RxUserData[31:0]	In	32-bit data received from the PMA

Timing Diagram

IP Initialization

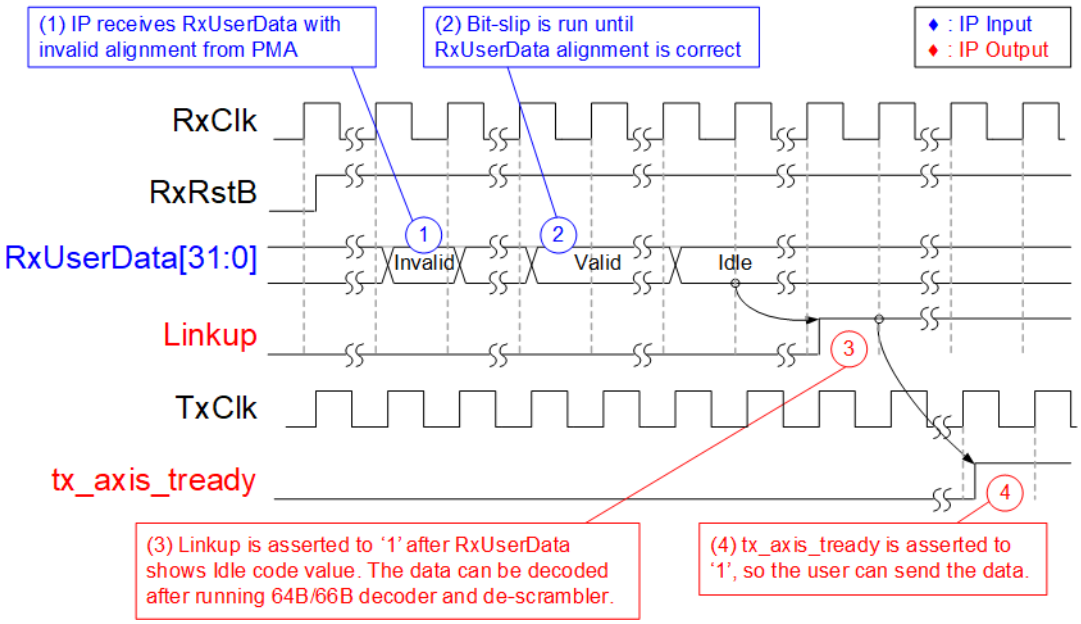

Figure 5: Rx Tuning and Linkup timing diagram

After RxRstB is de-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, tx_axis_tready is asserted to '1' in TxClk domain. More details of the initialization process are described as follows.

(1) IP receives RxUserData from the PMA every clock cycle, but the data alignment is still not corect.
(2) The received data is tuned until the data is aligned in valid format. The synchronization process is now completed.
(3) The received data is descrambled and 64B/66B decoded. After that, the controller waits until Idle code is found and then asserts Linkup signal to '1'.
(4) The IP is ready to transfer the data with user. tx_axis_tready is asserted to '1' for receiving the data from the user.

Transmit interface

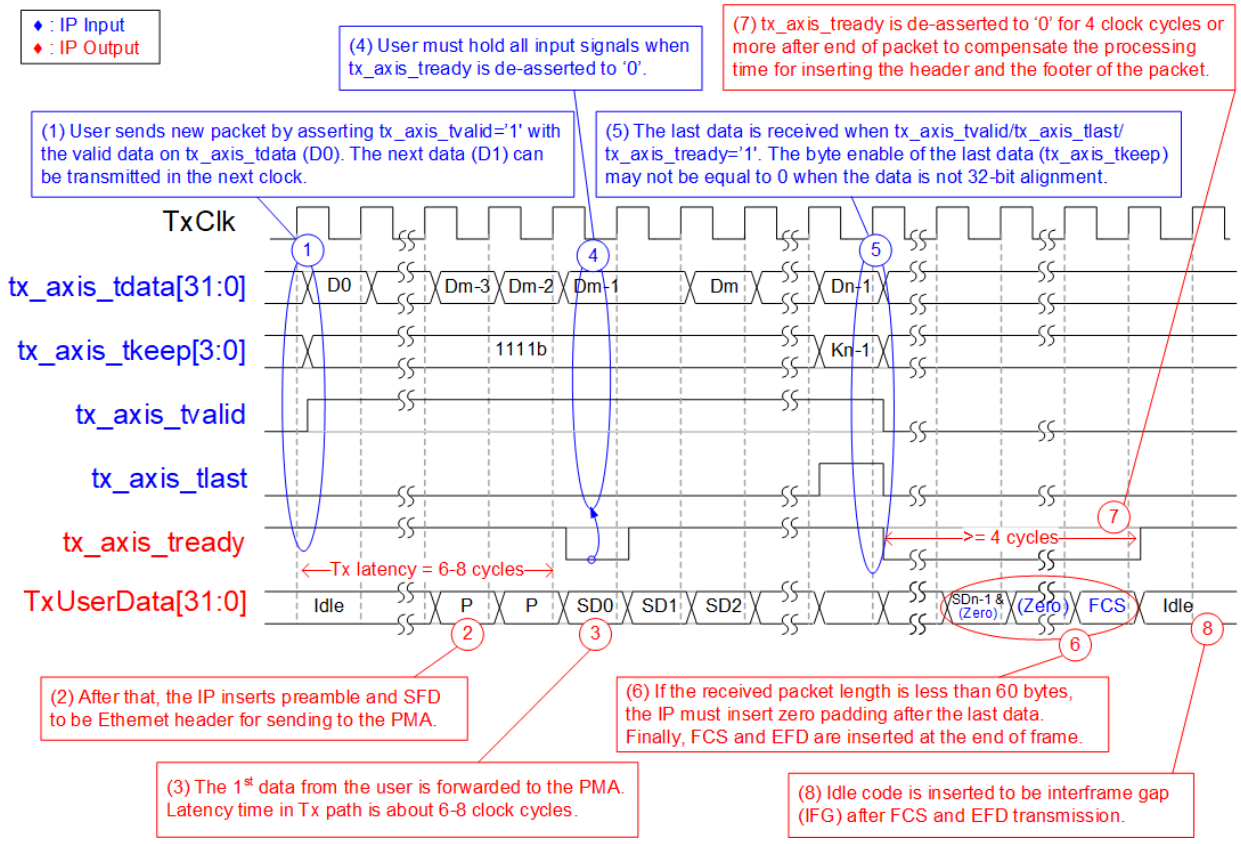

Figure 6: Transmit interface timing diagram

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. 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 cannot be decoded from 32-bit PMA data (tx_axis_tdata) without running data alignment, descrambler, and 64B/66B decoder.

(1) The new packet is found when tx_axis_tvalid is asserted to '1' with the IP ready status (tx_axis_tready='1'). Tkeep input must be always equal to Fh except the last data which may not equal to Fh for 32-bit unaligned data.
Note:Sometimes, tx_axis_tready can be de-asserted to '0' about 1-2 clock cycles after receiving the first data. So, the second data (D1) must hold the value until tx_axis_tready is re-asserted to '1'.
(2) After that, 7-byte preamble and SFD are sent to the PMA.
(3) The first data is sent by the IP to the PMA. Data latency of Tx path, measured by the first data on tx_axis_tdata and the first data on TxUserData, is about 6-8 clock cycles, depending on the data sequence in the Gearbox.
(4) When tx_axis_tready is de-asserted to '0', all input signals from user (tx_axis_tdata, tx_axis_tkeep, tx_axis_tvalid, and tx_axis_tlast) must hold the same value until tx_axis_tready is re-asserted to '1'. Typically, tx_axis_tready 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 (tx_axis_tvalid/tx_axis_tlast/tx_axis_tready ='1'), the IP ready (tx_axis_tready) is de-asserted to '0' to pause data transmission for waiting the IP to finish post-processing the packet. In this cycle, tx_axis_tkeep shows the byte enable of the last data which may be equal to 1111b (4-byte valid), 0111b (3-byte valid), 0011b (2-byte valid), or 0001b (1-byte valid).
(6) The last data which is encoded and encrypted by the IP is transmitted to the PMA. Zero-padding is inserted if the packet length is too short.
(7) After the last data is received from the user, tx_axis_tready 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 because the packet must be appended by the header and the footer. Therefore, tx_axis_tready is de-asserted to pause data input for several cycles during transferring packet header and packet footer.
(8) Idle code is always inserted at the end of packet to be the interframe gap (IFG) of the packet. The IP inserts at least 9-byte IFG for each packet.

Receive Interface

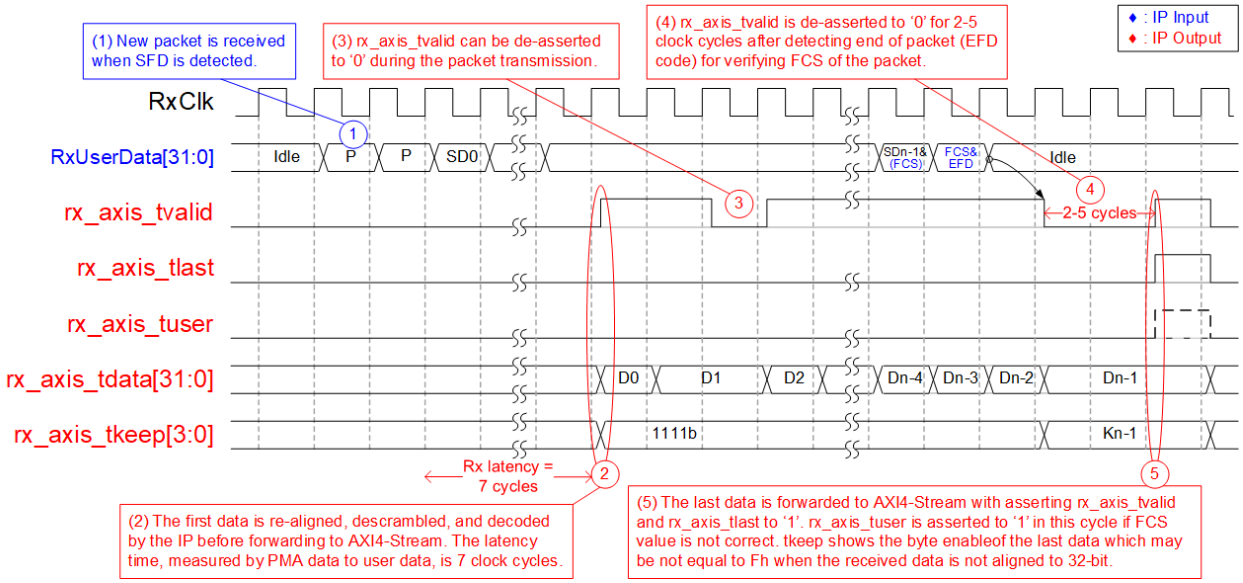

Figure 7: Receive interface timing diagram

When the new packet is received from the PMA, the IP re-align 32-bit data. After that, the data is descrambled and decoded 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 (rx_axis_tuser='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 is caused by the internal logic for converting PMA data to be user data. The data on the user interface is valid for all 32-bit except the last data which may be valid only some bytes. Therefore, rx_axis_tkeep is always equal to Fh, except the last data. rx_axis_tkeep of the last data may be equal to 0001b (1-byte valid), 0011b (2-byte valid), 0111b (3-byte valid), or 1111b (4-byte valid).
(3) The valid signal of user interface (rx_axis_tvalid) 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), rx_axis_tvalid is de-asserted to '0' for 2-5 clock cycles for verifying FCS of the received packet.
(5) The IP asserts rx_axis_tlast and rx_axis_tvalid to '1' with the last data on rx_axis_tdata. At the same time, rx_axis_tuser is de-asserted to '0' when the error is not detected in the received packet. Otherwise, rx_axis_tuser 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 ZCU102 evaluation board.

Recommended Design Experience

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

Ordering Information

This product is available directly from Design Gateway Co., Ltd. Please contact Design Gatway 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	21-May-2020	New release
1.1	29-Apr-2021	Update IP to version 2