LL10GEMAC IP Core Data Sheet

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

Core Facts

Provided with Core	
Documentation	Reference Design Manual Demo Instruction Manual
Design File Formats	Encrypted file
Instantiation Templates	VHDL
Reference Designs & Application Notes	Vivado Project, See Reference Design Manual
Additional Items	Demo on ZCU102, Alveo U50 and U250 cards
Support	
Support Provided by Design Gateway Co., Ltd.	

Features

- Support for 10G Ethernet MAC and PCS
- Direct connection with 32-bit PMA using Xilinx IP wizard.
- Low latency solution with a round-trip latency of 65.1 ns (18.6 ns for Tx path, 21.7 ns for Rx path, and 24.8 ns for PMA latency)
- AXI4-Stream interface for integration with user logic
- Minimal resource consumption
- Minimum Tx packet size of 5 bytes
- FCS (CRC-32) insertion and checking for data integrity
- 64B/66B Encoding and Decoding in compliance with IEEE802.3ae specification
- Support for the 10GBASE-R standard
- Zero padding appended for the Tx interface, while zero padding is not removed for the Rx interface
- Separate clock domains for transmit and receive interfaces operating at 322.265625 MHz
- Reference design available, including
  - A Loopback demo on the Xilinx development board (ZCU102)
  - An Accelerated Algorithmic Trading (AAT) demo on the Alveo accelerator cards (U50 and U250)

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

The low-latency network access has become crucial for various real-time applications, including High-Frequency Trading (HFT), Data Centers, and Real-Time Control Systems in industries such as Automotive and Industrial sectors. The Low-Latency 10 Gigabit Ethernet MAC (LL10GEMAC) provides an efficient and high-performance solution for low-latency networking using 10G Ethernet.

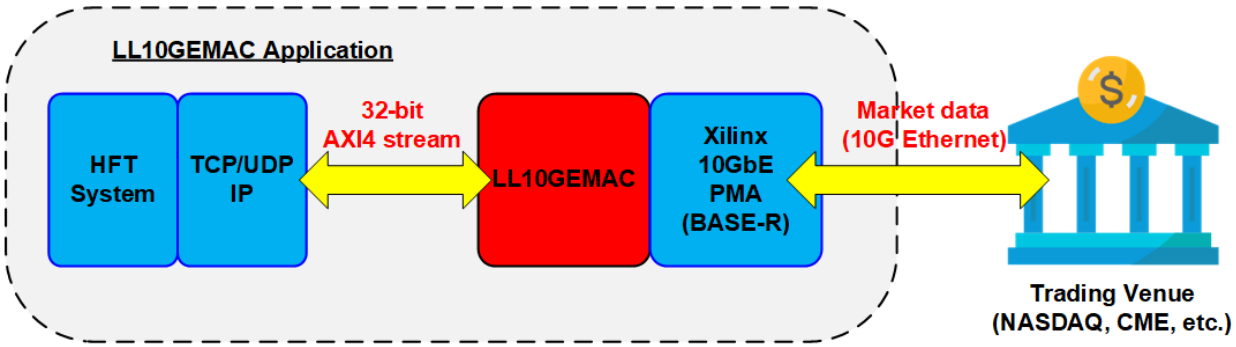

Figure 1: LLGEMAC Application

HFT involves executing high-speed trades with rapid data transmission and minimal response times. The LL10GEMAC IP core, implementing an Ethernet MAC and PCS, enables high-speed data transmission with minimal delay. Its user interface, utilizing a 32-bit AXI4 stream interface, allows for designing TCP and UDP engines using HLS, reducing development time. This combination empowers users to create efficient and optimized trading systems for HFT requirements.

Reference design

The LL10GEMAC IP provides two reference designs, the Loopback demo design on the Xilinx development board and the Accelerated Algorithm Trading (AAT) demo on Alveo accelerator cards.

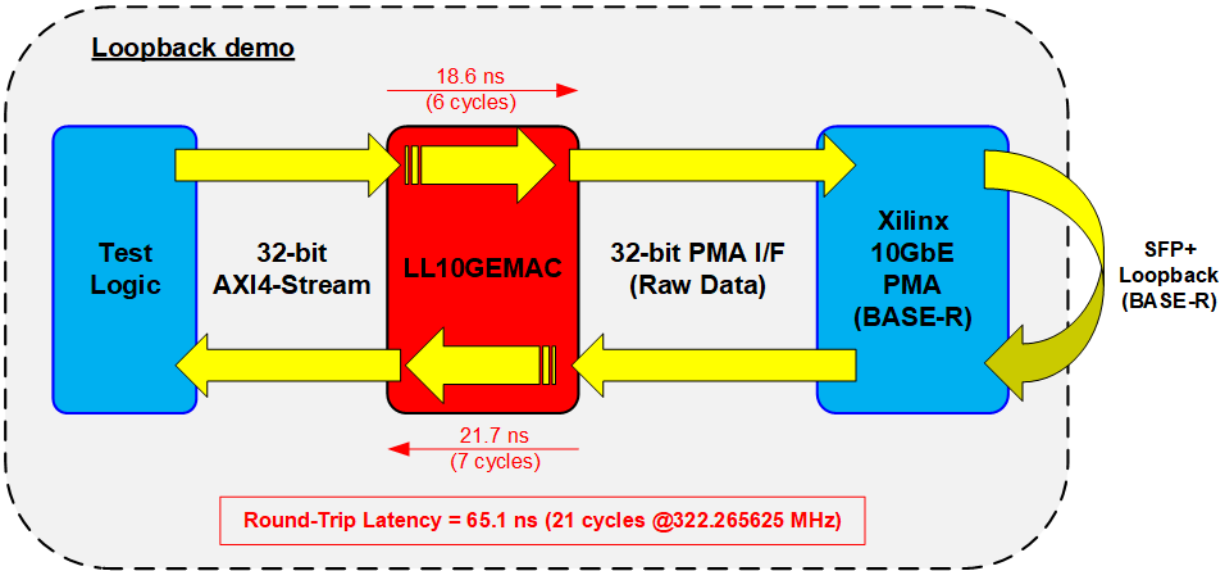

Figure 2: LL10GEMAC latency on Loopback demo

The Loopback demo provides a test logic to generate small packets and verify the return packets, measuring the round-trip latency time through the loopback logic. The round-trip latency measured from the Tx path to the Rx path of the LL10GEMAC's AXI4-Stream interface is 65.1 ns or 21 clock cycles at 322.265625 MHz, as shown in Figure 2.

Xilinx provides the Accelerated Algorithm Trading (AAT) demo, which includes the 10G/25G Ethernet Subsystem for Ethernet MAC, PCS, and PMA features. Other modules, such as TCP/IP engine, UDP/IP engine, Market data processing, Algorithm logic to generate the trading order, and Order generator, are designed using High-Level Synthesis (HLS) to complete the system. The AAT reference design is available for the Alveo accelerator cards.

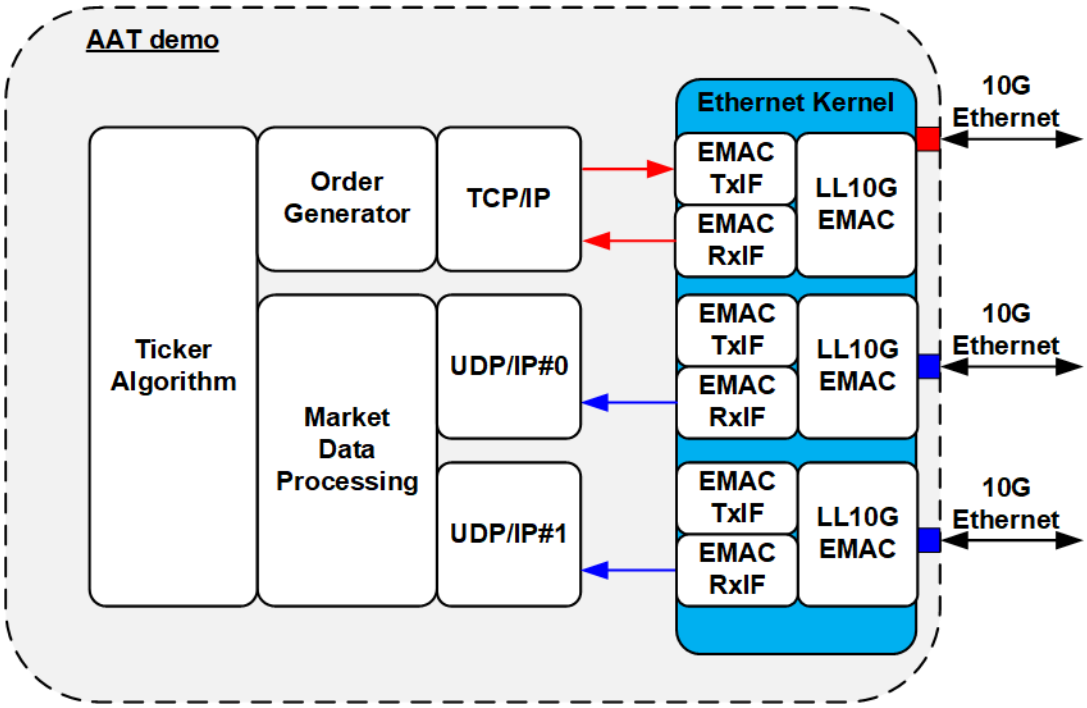

Figure 3: LL10GEMAC IP core using in AAT demo

In the modified AAT demo, the LL10GEMAC IP core replaces the 10G/25G Ethernet Subsystem inside the Ethernet Kernel, achieving lower latency with reduced resource consumption. For further details, please refer to the reference design document of the AAT demo utilizing the LL10GEMAC IP core.

During the execution of the loopback demo, the round-trip latency consists of both the latency time of the LL10GEMAC IP core and the Xilinx 10G PMA. The specific latency values of the PMA can be found in AR#68177 (https://www.xilinx.com/support/answers/68177.html). For the loopback demo, the UltraScale+ GTH Transceiver block is configured by the low-latency strategy, as indicated in Figure 4.

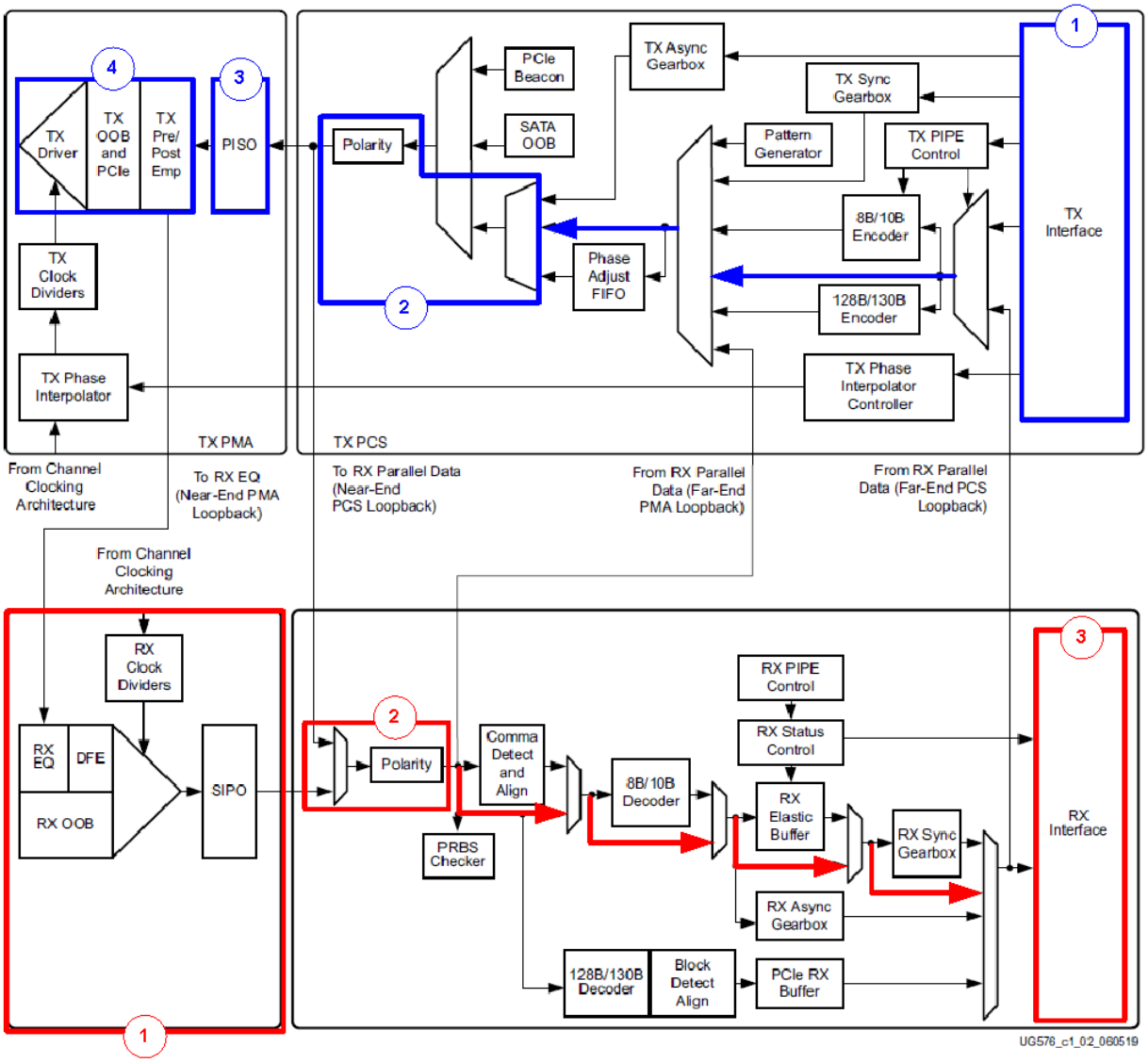

Figure 4: Xilinx PMA for 10G Ethernet by Raw Data (Ref-UG576 UltraScale GTH transceiver)

The latency value of PMA logic can be calculated as below.

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

The maximum 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

The maximum total Rx Latency is108.5 UI.

Therefore, the maximum PMA latency when running the loopback demo is 143 UI (Tx) + 108.5 UI (Rx). In the loopback demo on ZCU102 (UltraScale+ GTH transceiver) with a clock frequency of 322.265625 MHz for measuring latency, the total PMA latency corresponds to approximately 8 clock cycles or 24.8 ns.

Note: The transfer speed of 10G Ethernet is 10.3125 Gbps, so 1 UI is equivalent to 0.097 ns (1/10.3125G)

General Description

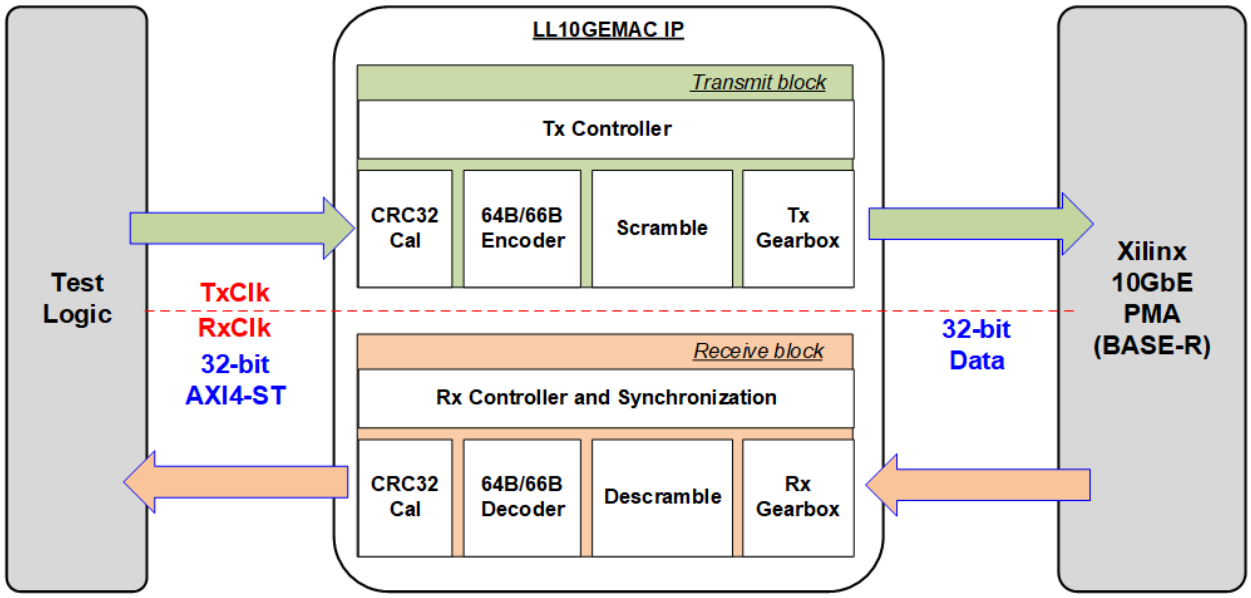

Figure 5: LL10GEMAC Block diagram

The LL10GEMAC IP core implements the MAC layer and the PCS (Physical Coding Sublayer) layer to provide a 10G Ethernet solution. Upon power up, the Rx controller and synchronization runs to calibrate the receive interface with the PMA (the transceiver) until a stable data lock is achieved. The received data from the PMA is continuous monitored to determine the link up status. Once the connection is established, the LL10GEMAC IP core asserts the ready signal to the user interface, allowing the user to transmit packets to the LL10GEMAC IP core.

During packet transmission, the LL10GEMAC IP core appends the preamble and SFD (Start frame delimiter) to create the packet header. At the end of packet, the LL10GEMAC IP core appends zero padding (for small packets), FCS (Frame check sequence), and IFG (Interframe gap) to form the packet footer. The packet is then encoded by 64B/66B encoder and scrambled by the Scramble block. The data is forwarded to the Tx Gearbox module to transmit 32-bit data to the PMA.

On the receiving end, the LL10GEMAC IP core first re-aligns the received packet using the Rx Gearbox. The data is then descrambled and decoded by the Descramble and 64B/66B decoder blocks, respectively. Finally, the FCS of the packet is verified and removed along with the preamble and SFD. Only the Ethernet data is forwarded to the AXI4 stream interface (AXI4-ST I/F). If the FCS is incorrect, the LL1GEMAC IP core asserts an error signal to AXI4-ST I/F.

During packet transmission, it is essential that the user always has the packet data ready until the end of the packet, with the data valid signal consistently asserted. However, the ready signal from the LL10GEMAC IP core may be de-asserted for one clock cycle every 32 clock cycles to align with the characteristics of the 64B/66B encoder.

Similarly, the user must be prepared to receive the packets from the LL10GEMAC IP core because there is no receive buffer within the IP core itself. During packet transmission, the data valid signal of the received packet is de-asserted for one clock cycle every 32 clock cycles to pause data transmission and align with the characteristics of the 64B/66B decoder.

Functional Description

LL10GMEAC IP facilitates bidirectional data transmission simultaneously, as shown in Figure 5. The Tx (transmit) and Rx (receive) logics operate independently in separate clock domains.

Transmit Block

Within the LL10GEMAC IP, the data packet received from the AXI4-ST interface is processed by CRC32 calculation (FCS), 64B/66B encoding, scrambling, and alignment before being forwarded to the PMA.

- Tx Controller

The Tx Controller adds the packet header and footer to the paket. The header consists of a 7-byte preamble and a 1-byte SFD (Start of frame delimiter), while the footer includes a 4-byte FCS (CRC-32) and 1-byte EFD (End of frame delimiter). If the packet is too short, zero-padding is appended after the last Ethernet data. After completing the transmission of each frame, an Idle is inserted as the IFG (Interframe Gap) for each packet. Additionally, the Tx Controller monitors the control signals on the AXI4-ST interface to detect new frame transmitted by the user. The ready signal on AXI4-ST is temporarily de-asserted to insert the header for 64B/66B encoding while the Gearbox operates.

- CRC32 Cal

This module calculates the 32-bit CRC of a data packet using a 32-bit data bus, which is input from the AXI4-stream interface. The FCS (CRC-32) is appended as the packet footer after transferring the Ethernet data, which may include zero-padding. The polynomial used for CRC-32 to generate the FCS follows the IEEE802.3ae standard.

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

- 64B/66B Encoder

64B/66B data encoding is employed in 10G Ethernet transmission to facilitate clock recovery in the Clock Data Recovery (CDR) module. The 64B/66B encoding logic is designed with minimal overhead and supports only 10GBASE-R as per the IEEE802.3ae specification. The module encodes the packet, including the header and footer, before passing it to the Scramble module.

- Scramble

Encoded data is scrambled to avoid long sequences of 1bs and 0bs. The scrambling polynomial specified by the IEEE802.3ae standard is used for this purpose.

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

- Tx Gearbox

To comply with the 64B/66B operation, each 64-bit data input from the user is encoded to 66-bit data. The output data size to the PMA is 32-bit. Therefore, the Tx Gearbox module realigns the encoded and scrambled data to 32-bit format. Since the bandwidth of the Scramble module is higher than that of PMA interface, the data from the user needs to be paused for one clock cycle every 32 clock cycles. During this pause time, the header of the 64B/66B encoder can be inserted.

Receive Block

The Receive Block consists of several submodules that handle the reception of data. It includes the reversed operation of the Transmit Block and incorporates a Synchronization block for additional functionality.

- Rx Gearbox

The Rx Gearbox submodule processes a 32-bit parallel data stream received from the PMA. The data stream is split into a 2-bit header and a 64-bit data without considering data alignment. To ensure the data in the correct order, a bit-slip logic is employed, realigning the data until the received data bus is correctly ordered. The control logic of the bit-slip function is designed in the subsequent module.

- Rx Controller and Synchronization

Following the completion of the reset sequence, the Synchronization submodule monitors the received data from the Rx Gearbox. Its purpose is to tune the data alignment by sending slip signal until the data can be correctly locked. Even after the Ethernet connection is established, continuous monitoring of data alignment occurs to return if any misalignment is detected.

Once the received packet from the Rx Gearbox is descrambled and decoded, the Start of Frame Delimiter (SFD) and Frame Check Sequence (FCS) of the packet are verified. If any errors are found in the received packet, the controller generates an error signal to the AXI4-ST interface. The header and footer are removed from the packet before forwarding it to the AXI4-ST interface. However, zero-padding withing the packet is not removed to minimize latency in the Rx path.

- Descramble

This submodule handles the descrambling of the data output from the Rx Gearbox before forwarding it to the 64B/66B Decoder.

- 64B/66B Decoder

The 64B/66B Decoder submodule decodes the descrambled data to identify the link up status, start of frame, and end of frame. The module provides outputs that include the data and the data type, which are monitored by the Rx controller to validate the data sequence within the packet.

- CRC32 Cal

  This submodule is identical to the CRC32 Cal module in the Transmit Block. However, in the Receiver Block, the calculated CRC32 is applied to verify the FCS extracted from the received packet. If the FCS is the received packet does not match the calculated CRC32, an error is asserted on the Rx AXI4-ST interface.

10GbE PMA (10GBASE-R)

The 10GBASE-R Physical Media Attachment (PMA), provided by Xilinx without the charge, is generated using the UltraScale FPGAs Transceivers Wizard. This wizard offers a template that assists users in configuring the transceiver parameters for 10GBASE-R operation. To ensure compatibility with LL10GEMAC, the following settings should be modified from the default value of the BASE-R template.

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

For more information about the Wizard, please refer to the following link.

https://www.xilinx.com/products/intellectual-property/ultrascale_transceivers_wizard.html

Core I/O Signals

Descriptions of all I/O signals are provided in Table 2.

Table 2: Core I/O Signals

Signal	Dir	Description
User Interface		
Linkup	Out	1b-Link up, 0b-Link down. Assert to 1b when the Ethernet connection is established and the PMA returns an Idle code in the 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=1b.
tx_axis_tkeep[3:0]	In	Byte enable of the 32-bit tx_axi_tdata. Each bit corresponds to the validity of a byte of tx_axis_tdata. Asserted to 1b when that byte is valid. Bit[0], [1], [2], and [3] correspond to tx_axis_tdata[7:0], [15:8], [23:16], and [31:24], respectively. When tx_axis_tvalid=1b, tx_axis_tkeep is equal to Fh for sending 32-bit data in each packet except the last data (tx_axis_tlast=1b). The byte enable of the last data can be equal to 1h, 3h, 7h, and Fh when 1 to 4 bytes of data are valid sequentially.
tx_axis_tvalid	In	Assert to 1b to transmit data. This signal must be continuously asserted to 1b from start of the packet to the end of packet. Note that the minimum size of transmitted data from the user is 5 bytes.
tx_axis_tlast	In	Assert to 1b to indicate the final word in the frame. Valid only when tx_axis_tvalid=1b.
tx_axis_tready	Out	A Handshaking signal indicated tx_axis_tdata has been accepted. Asserted to 1b when the data is completely received. When it is de-asserted, tx_axis_tdata/tkeep/tvalid/tlast must be latched to the same value until tx_axis_tready is re-asserted to 1b.
Rx AXI4 stream interface (Synchronous to RxClk)		
rx_axis_tdata[31:0]	Out	Received data. Valid when rx_axis_tvalid=1b.
rx_axis_tkeep[3:0]	Out	Received data byte enable. Each bit corresponds to the validity of a byte of rx_axis_tdata. Asserted to 1b when that byte is valid. Bit[0], [1], [2], and [3] correspond to rx_axis_tdata[7:0], [15:8], [23:16], and [31:24], respectively. The signal is valid when rx_axis_tvalid=1b.
rx_axis_tvalid	Out	Asserted to 1b when the received data is valid. During packet transmission, this signal may be de-asserted to 0b to pause data transmission.
rx_axis_tlast	Out	Assert to 1b to indicate the final word in the frame. Valid only when rx_axis_tvalid=1b.
rx_axis_tuser	Out	Valid at the end of the frame transmission (rx_axis_tlast=1b and rx_axis_tvalid=1b) to indicate if the frame contains an error. 0b: normal packet, 1b: error packet (SFD, FCS, or EFD is incorrect).
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

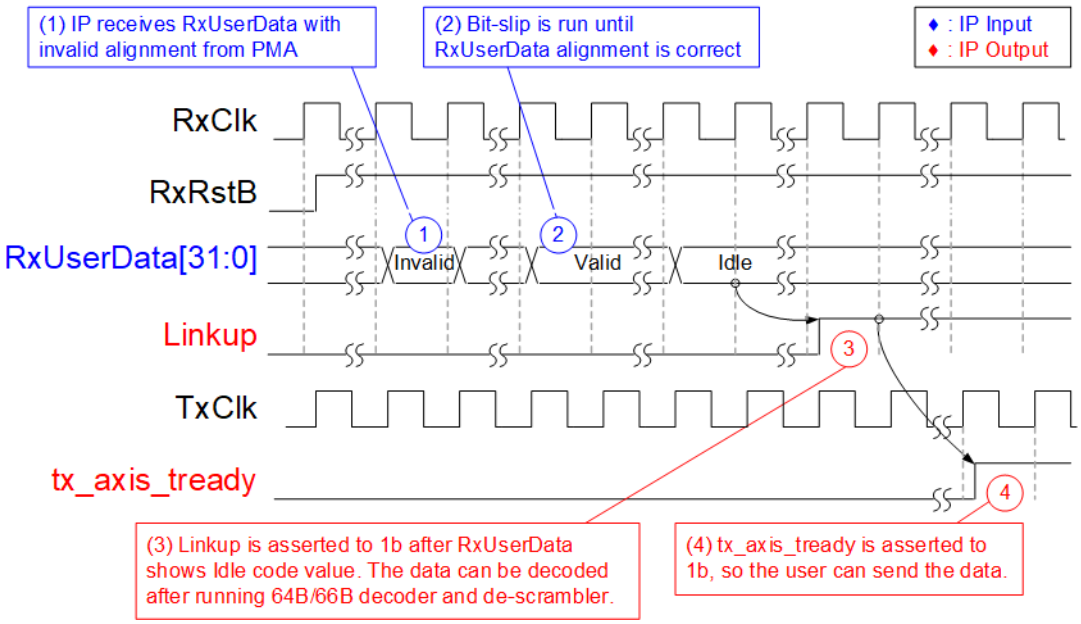

Figure 6: Rx Tuning and Linkup timing diagram

Upon de-assertion of RxRstB to 1b, the receive module inside the IP initiates the synchronization process to align and lock the received data from the PMA. Once the data is successfully locked and the Ethernet connection is established, the Linkup signal is asserted to 1b in the RxClk domain. Also, in the Tx clock domain, the tx_axis_tready is asserted to 1b. The initialization process can be further described as follows.

(1) The IP receives RxUserData from the PMA on every clock cycle, but the data alignment is initially incorrect.
(2) The received data undergoes tuning process until it is aligned a valid format. This synchronization process continues until completion.
(3) The aligned data is then descrambled and decoded using the 64B/66B decoding scheme. Subsequently, the controller waits for the detection of the Idle code. Once the Idle code is detected, the Linkup signal is asserted to 1b, indicating that the IP is ready for establishment of the Ethernet connection.
(4) The IP is prepared to received data from the user. To indicate its readiness to received data, the tx_axis_tready is asserted to 1b, enabling the IP to accept transmitted data from the user.

Transmit interface

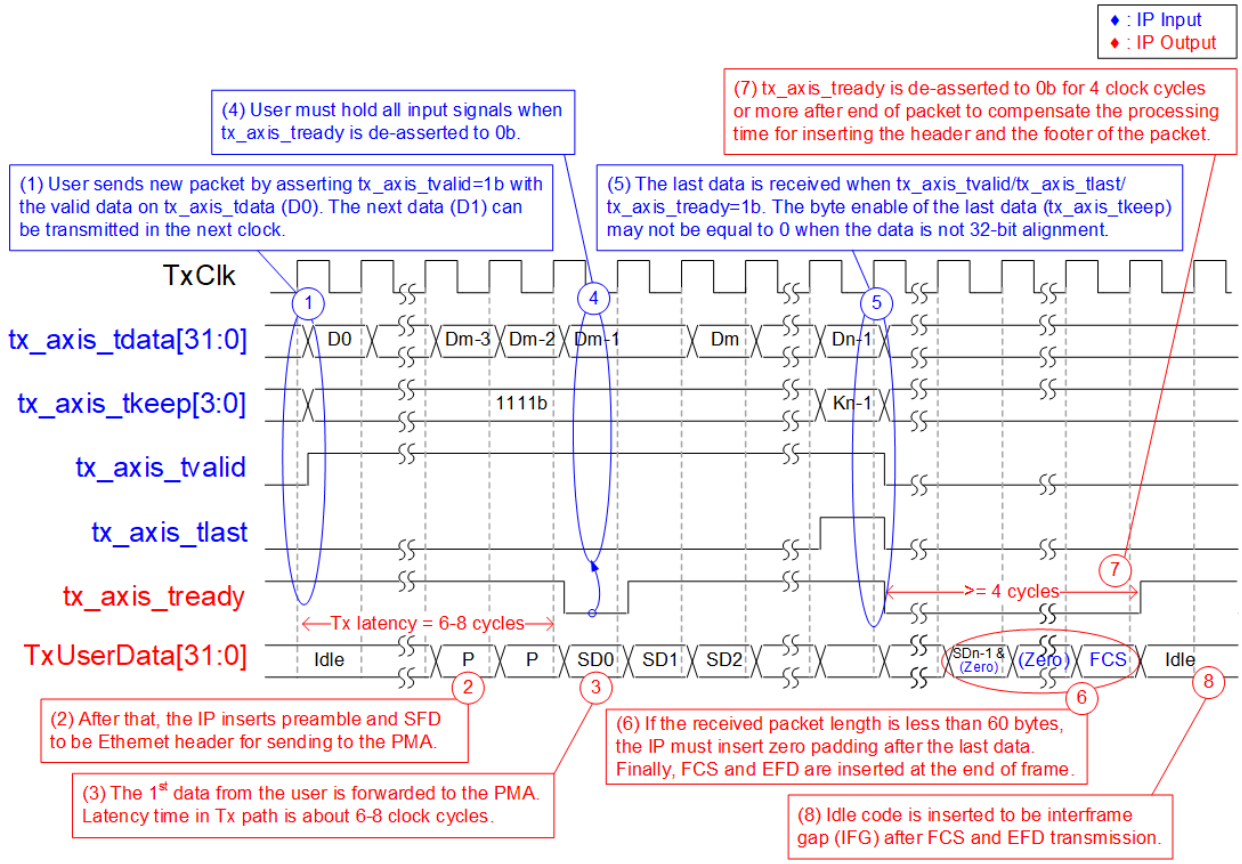

Figure 7: Transmit interface timing diagram

When a new frame is transmitted from the user, the IP performs several operations to prepare the packet for transmission. This includes inserting the header that are the Preamble and SFD, as well as the footer that are the FCS and EFD. All data, including the header and footer, generated by the IP undergoes encoding, scrambling, and re-alignment to a 32-bit format. In the case of short packets (less than 60bytes), zero-padding is added before transmitting the FCS.

The process of packet transmission can be described as follows.

(1) The IP identifies a new packet when tx_axis_tvalid is asserted to 1b while the IP is in a ready state (tx_axis_tready=1b). The tkeep input should always be equal to Fh, except for the last data in cases of unaligned 32-bit data.
Note: tx_axis_tready may be temporarity de-asserted to 0b for 1-2 clock cycles after receiving the first data. Therefore, the second data (D1) must retain its value until tx_axis_tready is re-asserted to 1b.
(2) The IP sends a 7-byte preamble and the SFD the PMA.
(3) The first data is transmitted from the IP to the PMA. The data latency of the Tx path, measured between the first data on tx_axis_tdata and the first data on TxUserData, typically ranges from 6 to 8 clock cycles, depending on the data sequence within the Gearbox.
(4) When tx_axis_tready is de-asserted to 0b, all input signals from user (tx_axis_tdata, tx_axis_tkeep, tx_axis_tvalid, and tx_axis_tlast) must maintain the same value until tx_axis_tready is re-asserted to 1b. Usually, tx_axis_tready is de-asserted to 0b for one cycle every 32 clock cycles to pause data transmission to add the header following the 64B/66B encoding process.
(5) Upon finding the last data is found (tx_axis_tvalid/tx_axis_tlast/tx_axis_tready =1b), the IP ready (tx_axis_tready) is de-asserted to 0b, pausing data transmission to allow the IP to complete post-processing of the packet. In this cycle, tx_axis_tkeep indicates 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 IP transmits the encoded and scrambled last data to the PMA, adding zero-padding if the packet length is too short.
(7) After receiving the last data from the user, tx_axis_tready remains de-asserted for at least 4 clock cycles. The number of output data from the IP exceeds the number of input data from the user due to the addition of the packet header and the footer. As a result, tx_axis_tready is temporarily de-asserted to pause data input for several cycles during the transmission of the packet header and footer.
(8) An Idle code is always inserted at the end of each packet, serving as the interframe gap (IFG). The IP ensures that at least 9-byte IFG is inserted for each packet.

Receive Interface

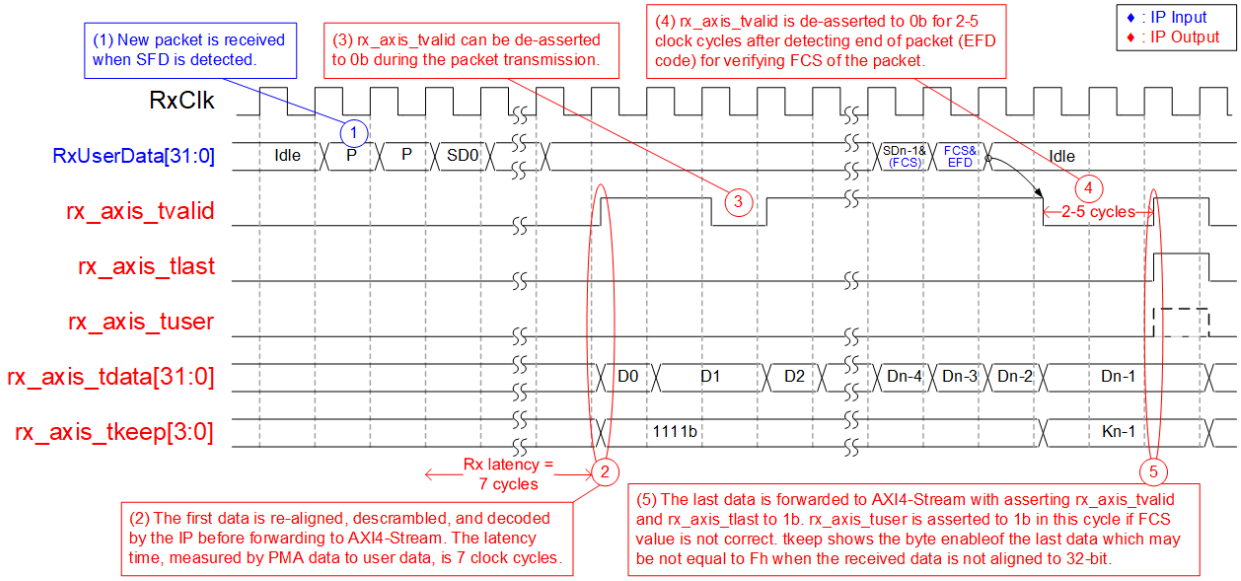

Figure 8: Receive interface timing diagram

When the IP receives a new packet from the PMA, it first re-aligns the 32-bit data and then descrambles and decodes it to check for the start and end of the packet. The header and footer of the packet are then verified and removed before forwarding it to the user. If the SFD, FCS, or EFD code is incorrect, an error signal (rx_axis_tuser=1b) is asserted to the user.

(1) The IP begins its operation after detecting the 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 user data. The data on the user interface is valid for all 32 bits except for the last data which may be valid only some bytes. Therefore, rx_axis_tkeep is always equal to Fh, except for 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) To remove the header from the data following 64B/66B decoding process, the valid signal of user interface (rx_axis_tvalid) is de-asserted to 0b for 1 clock cyle every 32 clock cycles.

(4) After detecting end of packet (EFD code), rx_axis_tvalid is de-asserted to 0b for 2-5 clock cycles to verify FCS of received packet.

(5) When there is no error detected in received packet, IP asserts rx_axis_tlast and rx_axis_tvalid to 1b with the last data on rx_axis_tdata. At the same time, rx_axis_tuser is de-asserted to 0b. If an error is detected, rx_axis_tuser is asserted to 1b at this clock cycle.

The IP removes packet header and footer from PMA before forwarding to 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, Alveo U50, and Alveo U250.

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 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
1.2	25-May-23	Add AAT demo on Alveo card
1.1	29-Apr-21	Update IP to version 2
1.0	21-May-20	New release