Design Gateway Co.,Ltd
54 BB Building 14th Fl., Room No.1402
Sukhumvit 21 Rd. (Asoke), Klongtoey-Nua,
Wattana, Bangkok 10110
Phone: 66(0)2-261-2277
Fax: 66(0)2-261-2290
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, using Synchronous Gearbox for low latency
- Low latency solution: 16 ns for Tx interface and 9.6 ns for Rx interface
- Minimum Tx packet size: 5 bytes
- Small resource utilization
- AXI4-stream interface with the user logic
- 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 312.5 MHz
- Reference design available on Xilinx development board (ZCU102)
- Customized service for following features.
  - Connection with Xilinx PMA IP in Asynchronous Gearbox for easily packet handling
  - Connection with Xilinx transceiver instead of Xilinx PMA IP for reducing the latency time

Table 1: Example Implementation Statistics

Family	Example Device	Fmax (MHz)	CLB Regs	CLB LUTs	CLB	IOB	BRAMTile	Design Tools
Kintex-Ultrascale	XCKU040FFVA1156-2E	312.5	883	1015	197	-	-	Vivado2019.1
Zynq-Ultrascale+	XCZU7EV-FFVC1156-2E	312.5	883	1015	202	-	-	Vivado2019.1
Virtex-Ultrascale+	XCVU9P-FLGA2104-2L	312.5	883	1016	204	-	-	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.

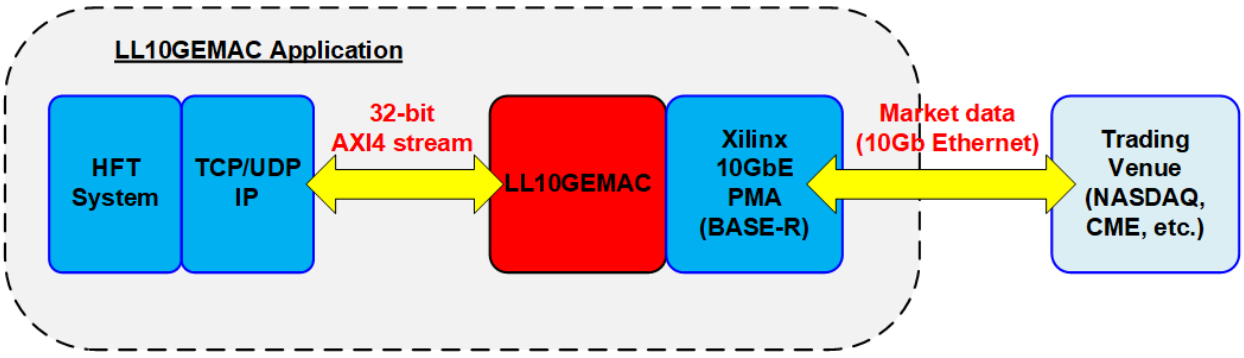

Figure 1: LLGEMAC Application

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 latency for both Tx and Rx interface.

- Tx latency is 16 ns (5 clock cycles of 312.5 MHz).
- Rx latency is 9.6 ns (3 clock cycles of 312.5 MHz).

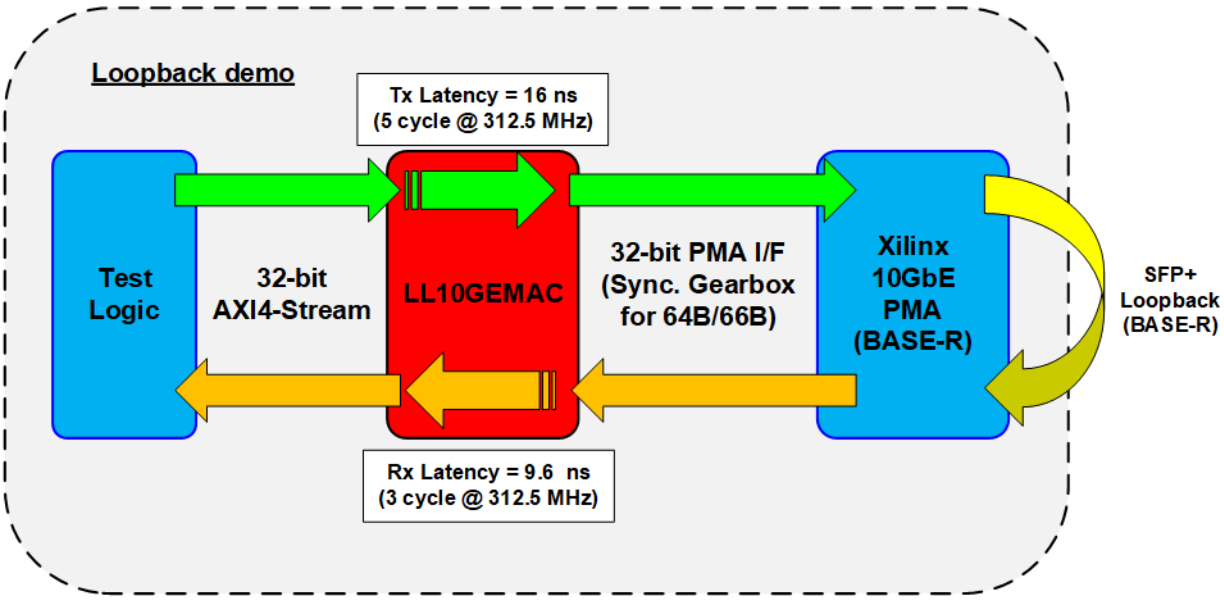

Figure 2: LL10GEMAC latency

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.

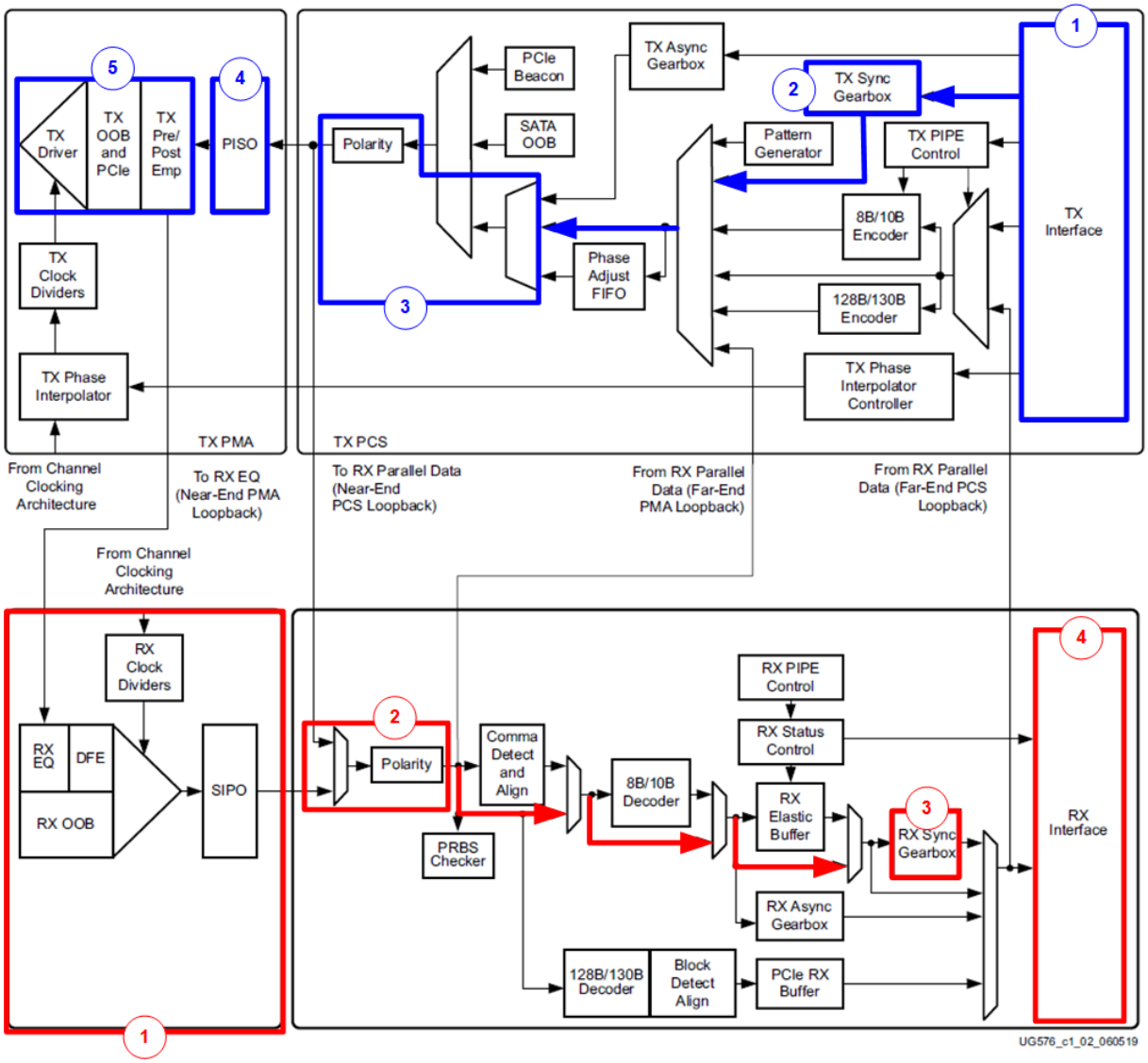

Figure 3: Xilinx PMA for 10GbE by Sync. Gearbox (Ref-UG576 UltraScale GTH transceiver)

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) Synchronous Gearbox = 64 – 128 UI
(3) To TX PCS/PMA boundary = 32 UI
(4) To Serializer = 64 UI
(5) PMA = 15 UI

Maximum value of total Tx Latency is 271 UI.

Rx Latency

(1) PMA = 60.5 UI
(2) PMA to PCS = 16 UI
(3) Synchronous Gearbox = 32 – 97 UI
(4) Rx Fabric Interface = 32 UI

Maximum value of total Rx Latency is 205.5 UI.

So, maximum latency time of PMA when running loopback demo is 271 + 205.5 UI. In our loopback demo by using 312.5 MHz clock for measuring latency time on ZCU102 (UltraScale+ GTH transceiver), the total latency time in PMA is about 13-15 clock cycles or 41.6 – 48 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

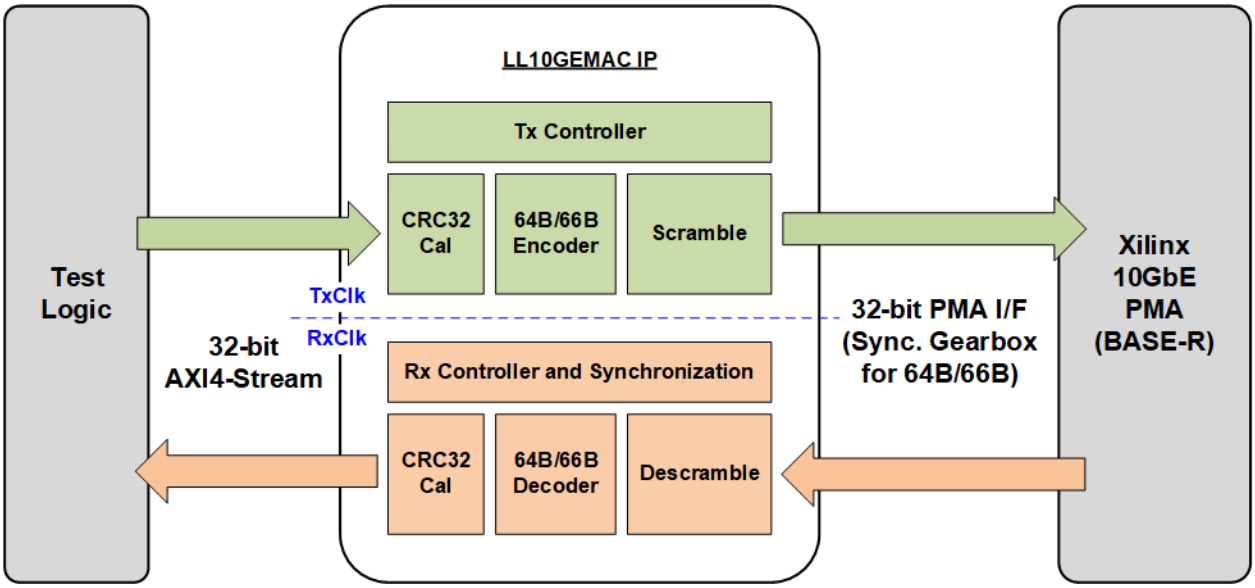

Figure 4: LL10GEMAC Block diagram

LL10GEMAC IP implements the MAC layer and the PCS (Physical Coding Sublayer) layer for 10Gb Ethernet solution. After power up, Rx interface of 10GbE PMA (Physical Medium Attachment) must be calibrated by Synchronization until the received data is locked. Next, Rx Controller monitors the received data from 10GbE PMA to check link up status of 10GbE. When the connection is ready, LL10GEMAC IP asserts ready signal to user interface. So, 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. Finally, the packet is encoded by 64B/66B Encoder and scrambled before transmitting to the PMA.

On the other hand, the received packet from the PMA is descrambled and decoded by LL10GEMAC IP. After that, 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 transmitting the header of 64B/66B to Sync.Gearbox module within PMA which must be sent for 2 clock cycles every 64 clock cycles.There is no transmit buffer inside LL10GEMAC IP to minimize latency time.

In the same way, the user must be always ready to receive the packet from LL10GEMACIP because there is no received buffer inside LL10GEMAC IP. The data valid of the received packet is de-asserted to ‘0’ for 2 clock cycles every 64 clock cycles to pause data transmission when receiving the header of 64B/66B.

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

- Tx Controller

The data packet from AXI4-ST is fed to LL10GEMAC IP for CRC32 calculation (FCS), 64B/66B Encoding and Scrambling before forwarding to the PMA. 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, 12-byte 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 and de-asserts ready signal on AXI4-ST when sending the header of 64B/66B, the packet header, the packet footer, IFG, and zero-padding.

- 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 including 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 which is the last process.

- Scramble

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

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

Received Block

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

- Rx Controller and Synchronization

After finishing the reset sequence, Synchronization monitors the received header and data bus for tuning data alignment 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 the PMA 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

The first process when receiving the data from the PMA is descrambling. 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 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 : Sync. gearbox for 64B/66B
- 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 received 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 to 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] for tdata[7:0], Bit[1] for tdata[15:8], and so on. When tx_axis_tvalid='1', tx_axis_tkeep = Fh for every clock except the last word which tx_axis_tlast='1'. The byte enable of the last word can be equal to 1h, 3h, 7h, and Fh for 1 – 4 byte 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] for tdata[7:0], Bit[1] for tdata[15:8], and so on. 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.

Signal	Dir	Description
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. 312.5 MHz for 32-bit interface.
GTTxSequence[6:0]	Out	Transmit sequence counter to the PMA.
GTTxHeader[5:0]	Out	Transmit header to the PMA.
GTTxData[31:0]	Out	Transmit 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 Tx interface. 312.5 MHz for 32-bit interface.
GTRxGearboxSlip	Out	Assert high to slip the gearbox contents to the next possible alignment.
GTRxHeaderValid[1:0]	In	Valid signal of GTRxHeader. Use only bit[0].
GTRxHeader[5:0]	In	Received header from the PMA. Use only bit[1:0] for 64B/66B. The header is valid when GTRxHeaderValid[0] is equal to '1'.
GTRxDataValid[1:0]	In	Valid signal of GTRxData. Use only bit[0].
GTRxData[31:0]	In	Received data from the PMA. Valid when GTRxDataValid='1'.

Note: More details of Tx PMA I/F and Rx PMA I/F are described in Xilinx Transceiver User Guide, depending on FPGA model. For example, GTH transceiver in UltraScale device document is "UG576 UltraScale Architecture GTH Transceivers User Guide".

https://www.xilinx.com/support/documentation/user_guides/ug576-ultrascale-gth-transceivers.pdf

Timing Diagram

Initialization

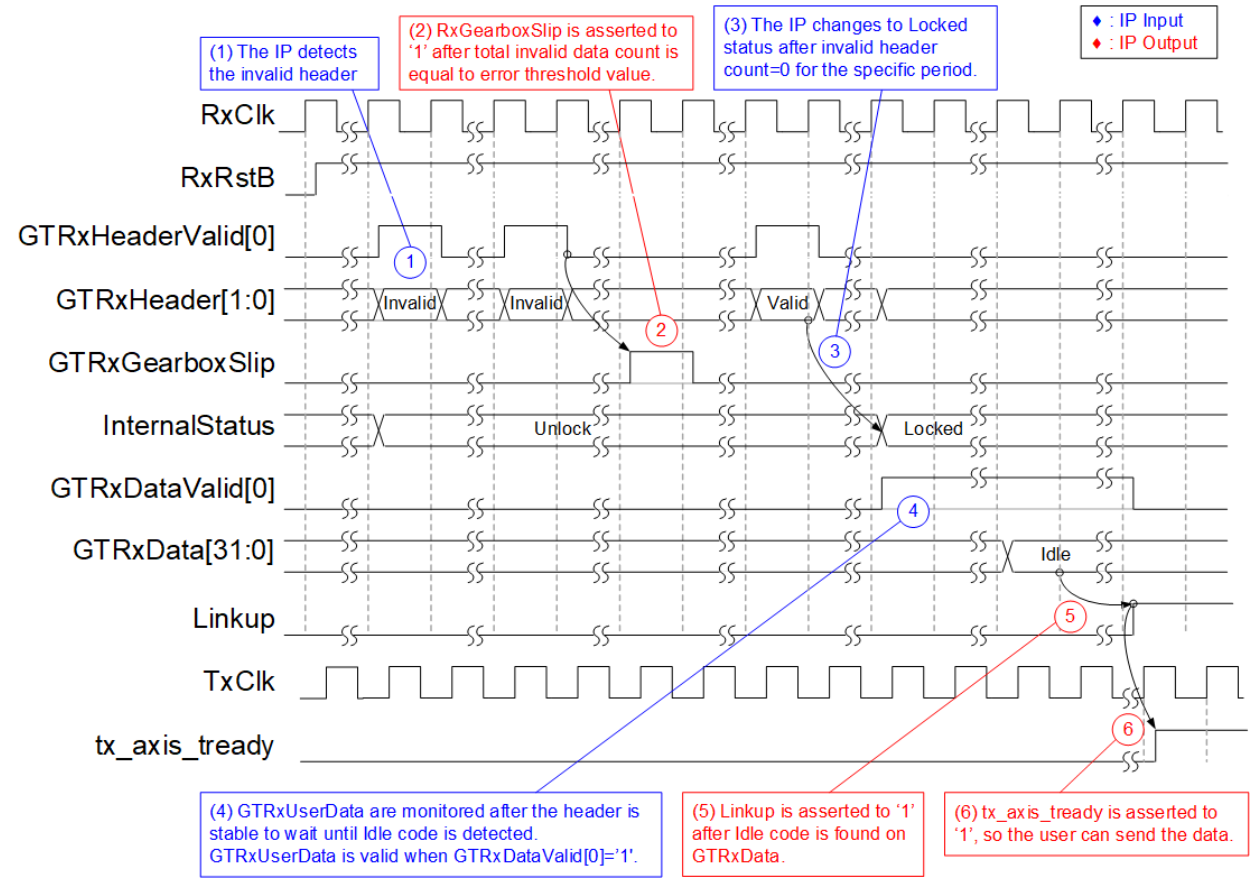

Figure 5: Rx Tuning and Linkup timing diagram

After RxRstB is de-asserted to '1', the received 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) When the header is invalid (GTRxHeader="00" or "11" when GTRxHeaderValid[0]='1'), the invalid counter inside the IP is increased.
(2) When the invalid counter is more than or equal to the invalid threshold value, the enable for data realignment (GTRxGearboxSlip) is asserted to '1'.
(3) Repeat step(1) – (2) until the invalid data is equal to 0 within the specified period. Now the synchronization operation is finished and the IP changes to locked status. After that, the monitoring module is still run as background process to check bit error rate. If bit error rate is more than the threshold value, the data realignment must be rerun.
(4) The data (GTRxData) and valid signal (GTRxDataValid[0]) are monitored to check the connection establishment. The data is valid when GTRxDataValid[0]='1'.
(5) After Idle code is detected on GTRxData and GTRxDataValid[0]='1', Linkup is asserted to '1'.
(6) The IP is ready to receive the data from user by asserting tx_axis_tready to '1'.

Transmit interface

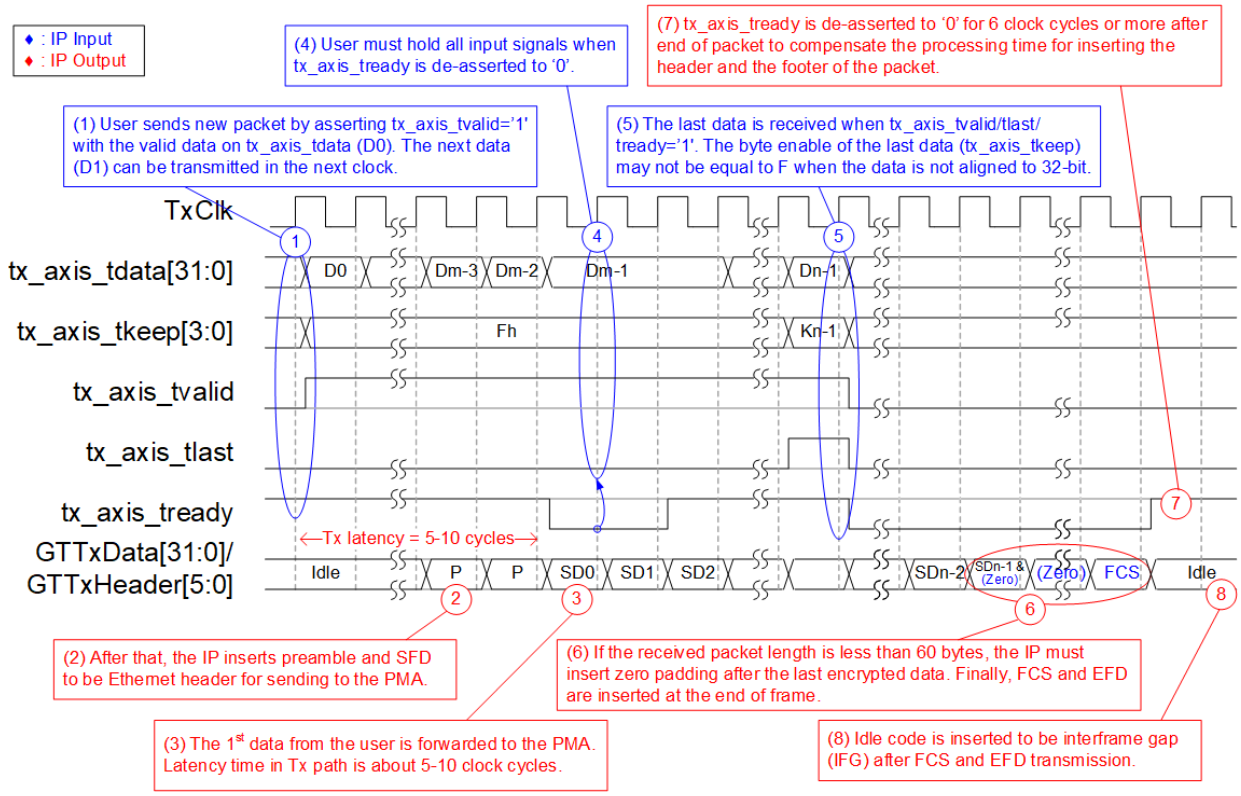

Figure 6: Transmit interface timing diagram

When the new frame is transmitted from the user, the IP inserts the header of the packet, i.e. 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 scrambled data which cannot be decoded without using descrambling logic. If the packet length is too short (less than 60-byte), zero-padding must be inserted before FCS transmission. Similar to other data, zero-padding is also scrambled before sending to the PMA.

(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: In some conditions, tx_axis_tready can be de-asserted to '0' in the next clock for 1-5 clock cycles. So, the next data (D1) must be holded the value until tx_axis_tready is re-asserted to '1'.
(2) After that, 7-byte preamble and SFD which are scrambled by the IP are sent to the PMA.
(3) The 1st data which is encoded and scrambled by the IP is sent to the PMA. Data latency is Tx path is about 5-10 clock cycles depending on the current data sequence on PMA interface.
(4) When tx_axis_tready is de-asserted to '0', all input signals from user (tx_axis_tdata, tkeep, tvalid, and 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 two cycle every 64 clock cycles to pause data transmission when running 64B/66B Synchronous Gearbox mode.

(5) After the last data is found (tx_axis_tvalid/tlast/tready='1'), the IP ready (tx_axis_tready) is de-asserted to '0' to pause data transmission for waiting the IP finishing the packet post-processing. In this cycle, tx_axis_tkeep shows the byte enable of the last data which may be equal to 0001b (1-byte), 0011b (2-byte), 0111b (3-byte), or 1111b (4-byte).
(6) The last data which is encoded and encrypted by the IP is transmitted to the PMA. If the last data is not aligned to 4-byte, FCS and EFD can be transmitted in the same clock cycle. Zero-padding is inserted when the packet is too short.
(7) After the last data is received from the user, tx_axis_tready is de-asserted at least 6 clock cycles. The number of data output from the IP is more than the number of data input from the user from the packet header and footer. So, tx_axis_tready is de-asserted to compensate the different value between the number of data input and data output.
(8) Idle code is always inserted at the end of packet to be the interframe gap (IFG) of the packet. The IFG inserted by the IP is about 9-17 bytes.

Received Interface

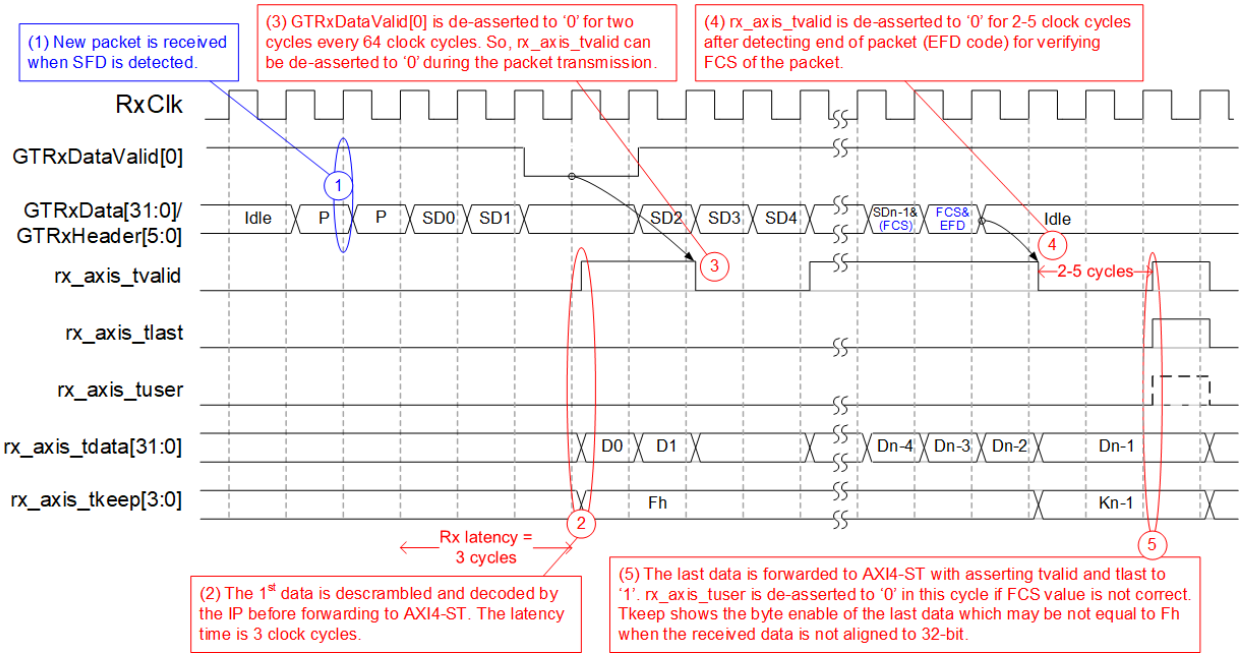

Figure 7: Received interface timing diagram

When the new packet is received from the PMA, the IP descrambles and decodes the data in 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 Rx PMA interface.
(2) After receiving the 1st data from the PMA for 3 clock cycles, the IP sends the 1st data to the user. The 1st data is descrambled and decoded by the IP. The data on the user interface is valid for all 32-bit except the last data which may be valid only some bytes. So, rx_axis_tkeep is always equal to Fh, except the last data which may be equal to 0001b (1-byte), 0011b (2-byte), 0111b (3-byte), or 1111b (4-byte).
(3) When the PMA de-asserts GTRxDataValid to '0' for 2 clock cycles every 64 clock cycles, the valid signal of user interface (rx_axis_tvalid) is de-asserted to '0' to pause data transmission. The latency time for deasserting the signal is 3 clock cycles, similar to the 1st data latency time.
(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/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	May-21-2020	New release