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

- Low-latency IP for receiving UDP packet streaming from 10G EMAC
- IP and UDP checksum calculation
- IPv4 protocol without IP fragmentation
- Operation mode: Unicast or Multicast (IGMPv2)
- Up to four sessions
- 3.2 ns latency time for receiving data
- 32-bit AXI4-stream interface for connecting with DG LL10GEMAC IP or Xilinx 10G/25Gb Ethernet Subsystem
- 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
  - Additional sessions
  - IGMPv3
  - IP fragmentation

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	1325	1332	314	-	-	Vivado2019.1
Zynq-Ultrascale+	XCZU9EG-FFVB1156-2-I	312.5	1325	1320	319	-	-	Vivado2019.1
Virtex-Ultrascale+	XCVU9P-FLGA2104-2L	312.5	1325	1320	343	-	-	Vivado2019.1

Applications

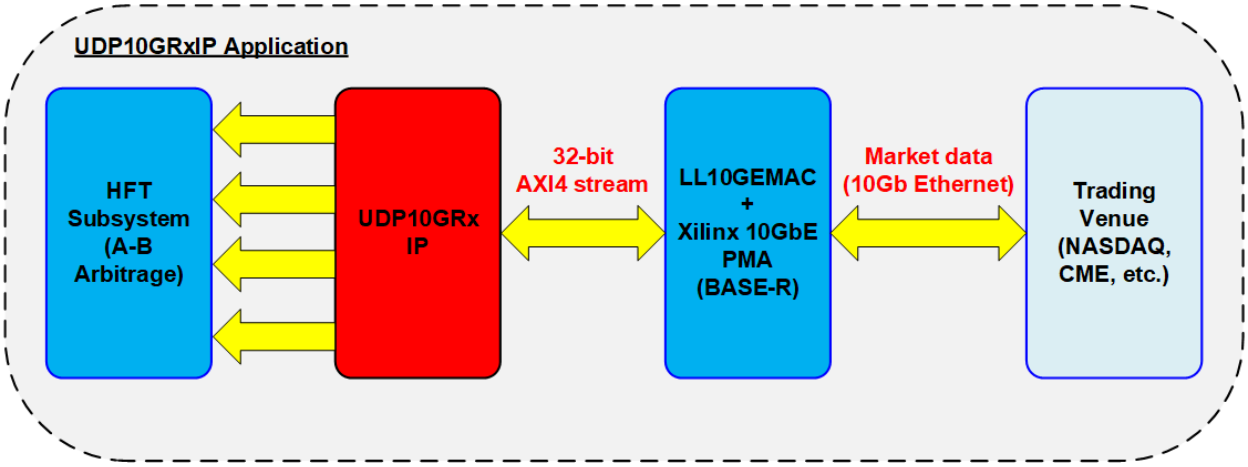

Figure 1: UDP10GRx IP Application

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 products. UDP10GRx IP is designed to receive UDP data stream via 10Gb Ethernet (BASE-R). Up to four sessions can be decoded by using one UDP10GRx IP. The example application of HFT for receiving market data by using UDP10GRx IP with LL10GEMAC is shown in Figure 1.

Reference design

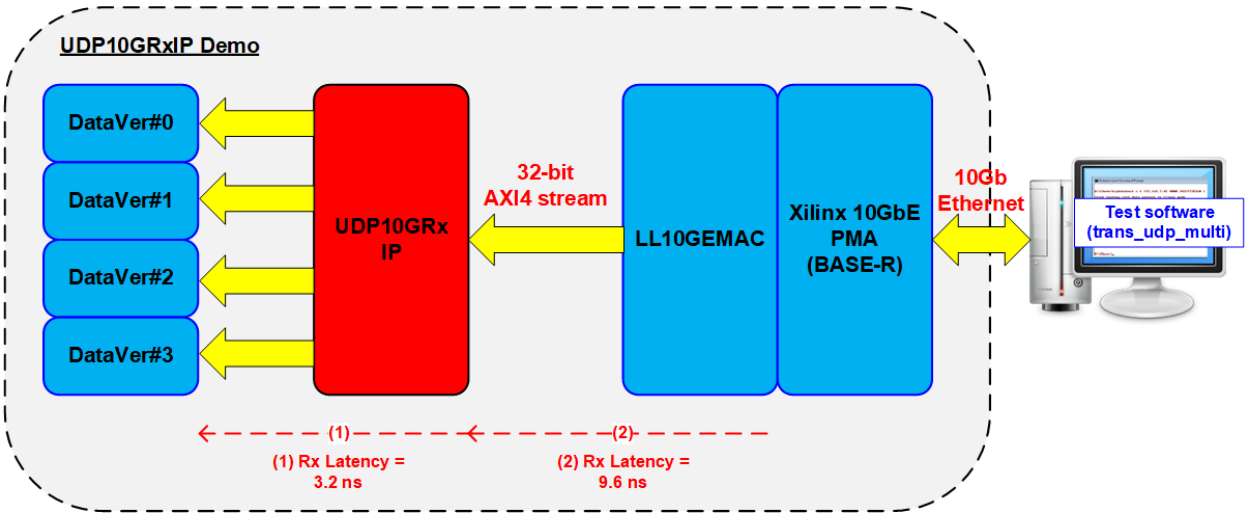

Figure 2: LL10GEMAC latency

UDP10GRx IP has been launched with the standard demo design on Xilinx devevelopment board. In the demo, the IP is connected with LL10GEMAC IP and Xilinx 10GbE PMA (BASE-R) for receiving UDP data transferred from test application running on Test PC. Up to four sessions can be transferred at the same time. So, the user interface of UDP10GRx IP is connected to the data verification logic for each session. Latency time for receiving data in UDP10GRx IP and LL10GEMAC is 3.2 ns and 9.6 ns, respectively.

General Description

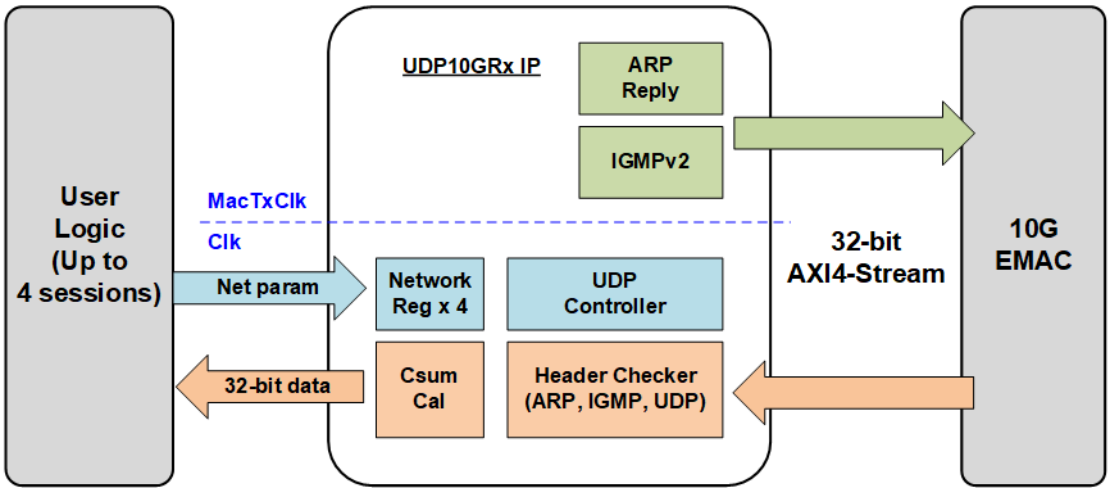

Figure 3: UDP10GRx IP Block diagram

UDP10GRx IP implements the logic to decode UDP data streaming in Multicast and Unicast mode. The user can set up to four network parameters for supporting receiving data from four sessions. Though four session parameters can be set, the IP address and MAC address of UDP10GRx IP are common parameters and must be the same value for all sessions.

The parameters input to UDP10GRx IP are split into two groups, i.e. common inputs and session inputs. Common inputs are the inputs which all sessions use such as operation mode which can be selected to be Unicast or Multicast. Session inputs are the inputs which are assigned by the different value for each session. The common inputs are loaded by UDP10GRx IP when the enable flag changes from all zero (all sessions are disabled) to be other value (some sessions are enabled). The session inputs of each session are loaded by UDP10GRx IP when the enable flag of the session changes from '0' to '1'. Multicast mode is applied for group communication while Unicast mode is applied for one-to-one communication. To change the operation mode between Multicast mode and Unicast mode, the user must disable all sessions by setting all zero and then enable some sessions by setting other values.

When running in Multicast mode, UDP10GRx IP sends Membership report for join group by using Multicast IP, defined in each session. After that, UDP10GRx IP decodes the UDP packet transferred by using Multicast IP and forwards to the user until the session is disabled. When the user disables the session, the message to leave group is sent by UDP10GRx IP. If the session is still active and Membership Query message is received, Membership report is returned by UDP10GRx IP.

When running in Unicast mode, UDP10GRx IP returns ARP reply after receiving ARP request which has the matched parameters with the active session. The UDP10GRx IP decodes the UDP packet transferred and extracts UDP data for sending to user until the session is disabled.

If some parameters in the IP header are not matched to the paramters of the active session, the packet is rejected by UDP10GRx IP. The packet is transferred to the user interface with asserting error flag when UDP checksum is error or EMAC asserts error flag at the end of Ethernet frame.

Functional Description

As shown in Figure 3, UDP10GRx IP uses different clock domain for interfacing with 10G EMAC in the different direction. Main clock is the clock for receiving the data from 10G EMAC and interfacing with the user interface, named Clk. Another is the clock which is used to be Tx interface of 10G EMAC, named MacTxClk.

Control block

- Network Reg

This module is designed to store the network paramters of all sessions from the user. The user can set the different parameters for four sesssions, except the IP address and MAC address of UDP10GRx IP which must be the same value for all sessions. The parameters of each session are loaded and latched when the user changes the enable flag of the session from ‘0’ to ‘1’.

- UDP Controller

The controller can run in two modes, i.e. Unicast mode and Multicast mode. Loading parameter is operated when the enable flag is changed, so the controller is always monitored enable flag of each sesssion. When the session is active, the parameters of the session are loaded to verify by Header Checker for decoding the packet which has matched parameters. Also, the controller starts ARP Reply or IGMPv2 module to create the packet during session initialization process or acknowledge packet must be returned by the IP.

Transmit block

- ARP Reply

This module is run on MacTxClk domain for creating ARP reply packet following the parameters defined by UDP controller. This module is enabled when running in Unicast mode only.

- IGMPv2

This module is run on MacTxClk domain for creating Membership report message to join group or leave group message following IGMPv2 protocol when UDP controller sends the request to generate the message. This module is enabled when running in Multicast mode only.

Received Block

- Header Checker

This module is designed to verify the header of the packet. Four parameter sets for comparing four sessions are applied to check the header of the received packet at the same time when the user enable four sessions. UDP data is extracted from the UDP packet and forwarded to the user interface if the parameters in the header is matched to the active session. ARP request is supported in Unicast mode while Membership Query (IGMPv2) is supported in Multicast mode only.

- Csum Cal

This module is designed to calculate IP checksum and UDP checksum of the received packet. When IP checksum is not correct, the packet is rejected. If UDP checksum is not correct, the packet is forwarded to the customer with asserting error flag.

10 Gb Ethernet MAC

To achieve the lowest latency, it is recommended to use LL10GEMAC IP from DesignGateway for connecting with UDP10GRx IP. More details of LL10GEMAC IP are described in following website.

https://dgway.com/Lowlatency-IP_X_E.html

Another solution of 10Gb Ethernet MAC for connecting with UDP10GRx IP is 10G/25G Ethernet Subsystem, provided by Xilinx. The user must select the data bus to be 32-bit interface for the low latency solution. More details of the IP are described in following website.

https://www.xilinx.com/products/intellectual-property/ef-di-25gemac.html

User Logic

The user interface of UDP10GRx IP is divided to two groups, i.e. network parameters and data stream interface. The network parameters can be assigned by using simple registers or constant value to be the input of the IP. Data stream interface for receiving data of four sesssions is designed by 32-bit shared data bus with the individual valid signal for each session. The user can forward the valid signal and the data bus to be the write enable and the write data of the internal data buffer respectively.

Core I/O Signals

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

Table 2: Core I/O Signals

Signal	Dir	Description
General interface (Synchronous to Clk)		
IPVersion[31:0]	Out	IP version number
TestPin[31:0]	Out	Reserved to be IP Test point.
RstB	In	Synchronous reset signal to IPcore. Asserted to '0' to reset the IP.
Clk	In	Clock domain which is synchronous to received interface of EMAC. When running 10Gb Ethernet with low latency mode, this clock is equal to 312.5 MHz.
User control interface (Synchronous to Clk)		
McastEn	In	IP mode. '0'-Unicast mode, '1'-Multicast mode All sessions must operate in the same mode. *Note: The IP loads McastEn, SrcMacAddr, and SrcIPAddr when SSEnable changes from 0000b to other values (some sessions have been active).*
SrcMacAddr[47:0]	In	MAC address of UDP10GRx IP. Use the same value for all sessions.
SrcIPAddr[31:0]	In	IP address of UDP10GRx IP. Use the same value for all sessions.
SSEnable[3:0]	In	Session enable set by the user. '0'-Disable, '1'-Enable. Each bit is applied to enable each session. Bit[0]-Session#0, [1]-Session#1, and so on. *Note: Before asserting SSEnable to '1', SSActive of that session must be equal to '0' to confirm that the session is inactive.*
SSActive[3:0]	Out	Session status returned by the IP. '0'-Inactive, '1'-Active. Each bit is referred to each session. Bit[0]-Session#0, [1]-Session#1, and so on.
McastIPAddr0-3[31:0]	In	Multicast IP address for each session. Use only when the IP runs in Multicast mode. *Note: The IP loads the parameters for each session when SSEnable of that session changes from '0' to '1'. The number after the parameter name is the session number.* *For example,* *Session#0: Use McastIPAddr0, SrcPort0, DstIPAddr0, and DstPort0.* *Session#1: Use McastIPAddr1, SrcPort1, DstIPAddr1, and DstPort1.*
SrcPort0-3[15:0]	In	Port number of UDP10GRx IP for receiving data from each session. This value is the port number of the receiver in the received packet.
DstIPAddr0-3[31:0]	In	Target IP address of each session. This value is the IP address of the sender in the received packet.
DstPort0-3[15:0]	In	Target port number of each session. Use only when the IP runs in Unicast mode. This value is the port number of the sender in the received packet.

Signal	Dir	Description
User data interface (Synchronous to Clk)		
UDPRxValid[3:0]	Out	Asserted to '1' when UDPRxData is valid. Four bits are applied to be valid of each session. Only one bit can be asserted to '1' at a time. Bit[0]-Session#0, [1]-Session#1, and so on.
UDPRxData[31:0]	Out	UDP data output, shared signal for all sessions. Valid when UDPRxValid[x]='1'.
UDPRxByteEn[3:0]	Out	Byte enable of UDPRxData, shared signal for all sessions. Valid when UDPRxValid[x]='1'. During packet transmission, this signal is equal to 1111b for enabling all 32-bit data except the last data of the packet (UDPRxEOP='1') which can be equal to four values, i.e. 0001b (Byte#0 valid), 0011b (Byte#0-#1 valid), 0111b (Byte#0-#2 valid), and 111b (all bytes valid).
UDPRxSOP	Out	Asserted to '1' when sending the first data of the packet, shared signal for all sessions. Valid when UDPRxValid[x]='1'.
UDPRxEOP	Out	Asserted to '1' when sending the last data of the packet, shared signal for all sessions. Valid when UDPRxValid[x]='1'.
UDPRxError[7:0]	Out	Some bits are asserted to '1' when the received packet has error, shared signal for all sessions. Valid when UDPRxValid[x]='1' and UDPRxEOP='1'. [0]-UDP checksum of the packet is error. [1]-EMAC asserts Error at the end of packet. [7:2]-Reserved
Rx EMAC I/F (Synchronous to Clk)		
MacRxData[31:0]	In	Received data. Valid when MacRxValid='1'.
MacRxValid	In	Asserted to '1' when MacRxData is valid.
MacRxEOP	In	Asserted to '1' when receiving the last data of the packet.
MacRxError	In	Asserted to '1' when the packet has error. Valid when MacRxValid='1' and MacRxEOP='1'.
Tx EMAC I/F (Synchronous to MacTxClk)		
MacTxClk	In	Clock signal for Tx interface of EMAC IP.
MacTxData[31:0]	Out	Transmitted data. Valid when MacTxValid='1'.
MacTxValid	Out	Asserted to '1' when MacTxData is valid.
MacTxByteEn[3:0]	Out	Byte enable of MacTxData. Four bits are applied for 32-bit data bus. Similar to UDPRxByteEn, this signal is equal to 1111b for enabling all 32-bit data in one packet except the last data of the packet (MacTxEOP='1') which can be equal to four values.
MacTxSOP	Out	Asserted to '1' when transmitting the first data of the packet.
MacTxEOP	Out	Asserted to '1' when transmitting the last data of the packet.
MacTxReady	In	Asserted to '1' by EMAC IP when the transmitted data is received correctly. If MacTxReady='0', transmitted data and control signals (MacTxData, MacTxValid, MacTxByteEn, MacTxSOP, and MacTxEOP) must hold the same value until MacTxReady is re-asserted to '1'.

Timing Diagram

IP Reset

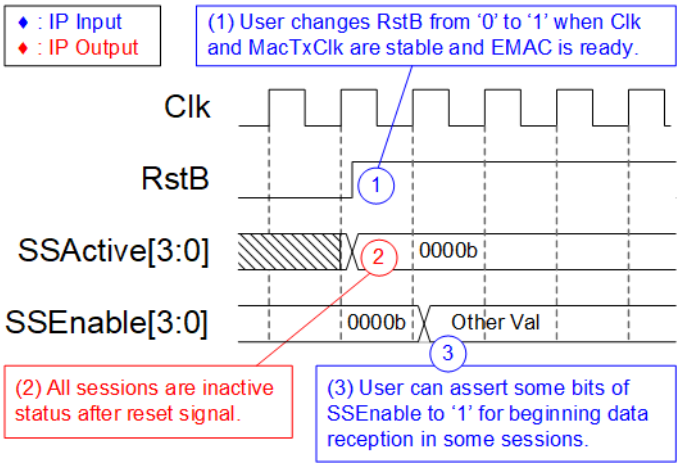

Figure 4: IP Reset process

(1) RstB is de-asserted to '1' by user logic after both clock inputs to UDP10GRx IP (Clk and MacTxClk) are stable. Also, EMAC should be ready status for transferring data with the IP.
(2) SSActive, output from UDP10GRx IP, is equal to 0h after reset process to show all sessions are inactive.
(3) The IP is ready status, so user can assert some bits of SSEnable to '1' to enable data reception from EMAC by using the active session parameters.

IP Initialization

The parameters which are the input signals of User control interface are split into two groups, i.e. common parameters which are applied for all sessions and session parameters which are independently defined for each session. Timing diagram for loading common parameters are shown in Figure 5.

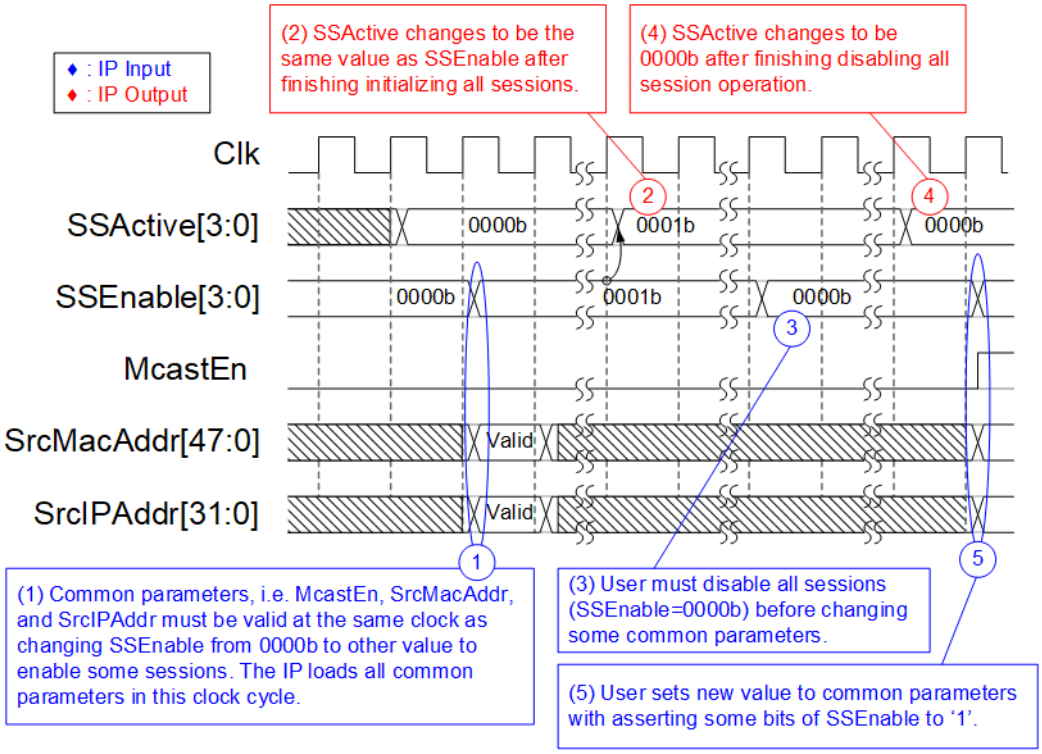

Figure 5: Initialization for common parameter

(1) When user begins the IP operations by changing SSEnable from 0000b to other value (asserting some bits to '1' to enable some sessions), all common parameters (McastEn, SrcMacAddr, and SrcIPAddr) must be valid at the same clock. After that, the IP begins initializing all active sessions by using the common parameters and the session parameters.
(2) After finishing to initialize all active sessions, SSActive value are equal to SSEnable. Now the IP is ready to receive the data which has the parameters matching to the active session. UDP data of the received packet which is valid is extracted and forwarded to the user.
(3) When the user finishes the operation or some common parameters must be changed, SSEnable must be de-asserted to 0000b to disable all sessions.
(4) SSActive changes to 0000b after finishing to disable all active sessions.
(5) The user can set the new value to common parameters and assert some bits of SSEnable to '1' for beginning the IP initialization by using the new parameters.

Session Initialization (Multicast mode)

The session parameters are loaded to the IP when SSEnable of the session changes from '0' to '1'. Figure 6 shows the example when enabling two sessions, i.e. session#0 and #2 in Multicast mode.

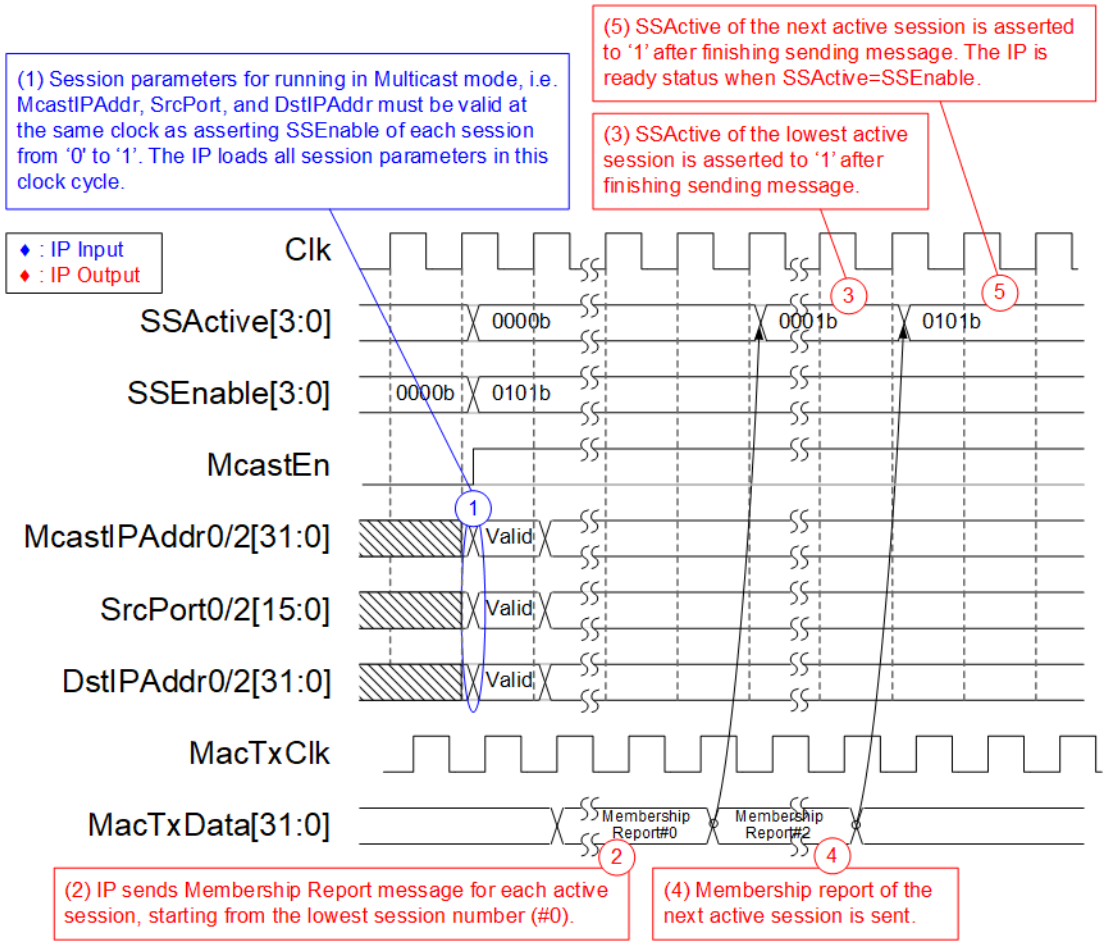

Figure 6: Multicast initialization

(1) When some bits of SSEnable change from '0' to '1', the session parameters of the enabled session are loaded to the IP in the same clock. In Multicast mode, three parameters are applied for each session, i.e. McastIPAddr, SrcPort, and DstIPAddr.

(2) IP begins the session initialization in Multicast mode by sending Membership report following the parameters of the active session. The IP initializes one session at a time, starting from the lowest session number. In the example, session#0 is initialized firstly.

(3) After finishing sending Membership report, SSActive of the completely initialized session (session#0) is asserted to '1'.

(4) If some sessions still not be initialized, Membership report of the next session number (session#2) is sent.

(5) SSActive of the next session (session#2) is asserted to '1'. Step 4) – 5) is repeated until SSActive is equal to SSEnable. After that, the IP is ready to receive UDP packet which has the matched parameters with the active session.

Session Initialization (Unicast mode)

Similar to Multicast mode, the session parameters are loaded to the IP when SSEnable of the session changes from '0' to '1'. Figure 7 shows the example when two sessions, i.e. session#1 and #3 are enabled in Unicast mode.

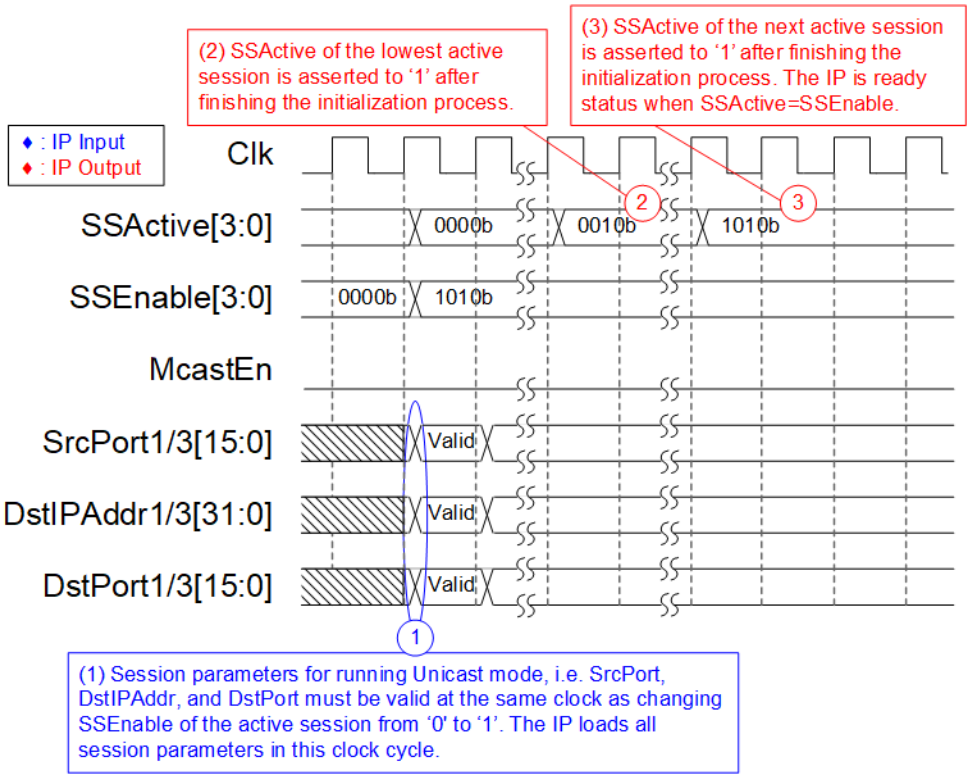

Figure 7: Unicast initialization

(1) When some bits of SSEnable change from '0' to '1', the session parameters of the enabled session are loaded to the IP in the same clock. In Unicast mode, three parameters are applied for each session, i.e. SrcPort, DstIPAddr, and DstPort.

(2) IP begins the session initialization process. Similar to Multicast mode, the IP initializes one session at a time, starting from the lowest session number. In the example, session#1 is initialized firstly. After that, SSActive of the completely initialized session (session#1) is asserted to '1'.

(3) The initialization of the next session number is run until SSAcive is equal to SSEnable. After finishing the initialization process, the IP is ready to decode UDP packet which has the matched parameters with the active session.

User Data Interface

When the IP receives UDP packet from EMAC, the parameters in the header are verified. The packet is decoded and only UDP data is forwarded to the user when the parameters are matched with some active session parameters. The IP asserts error to the user if checksum in UDP packet is not correct.

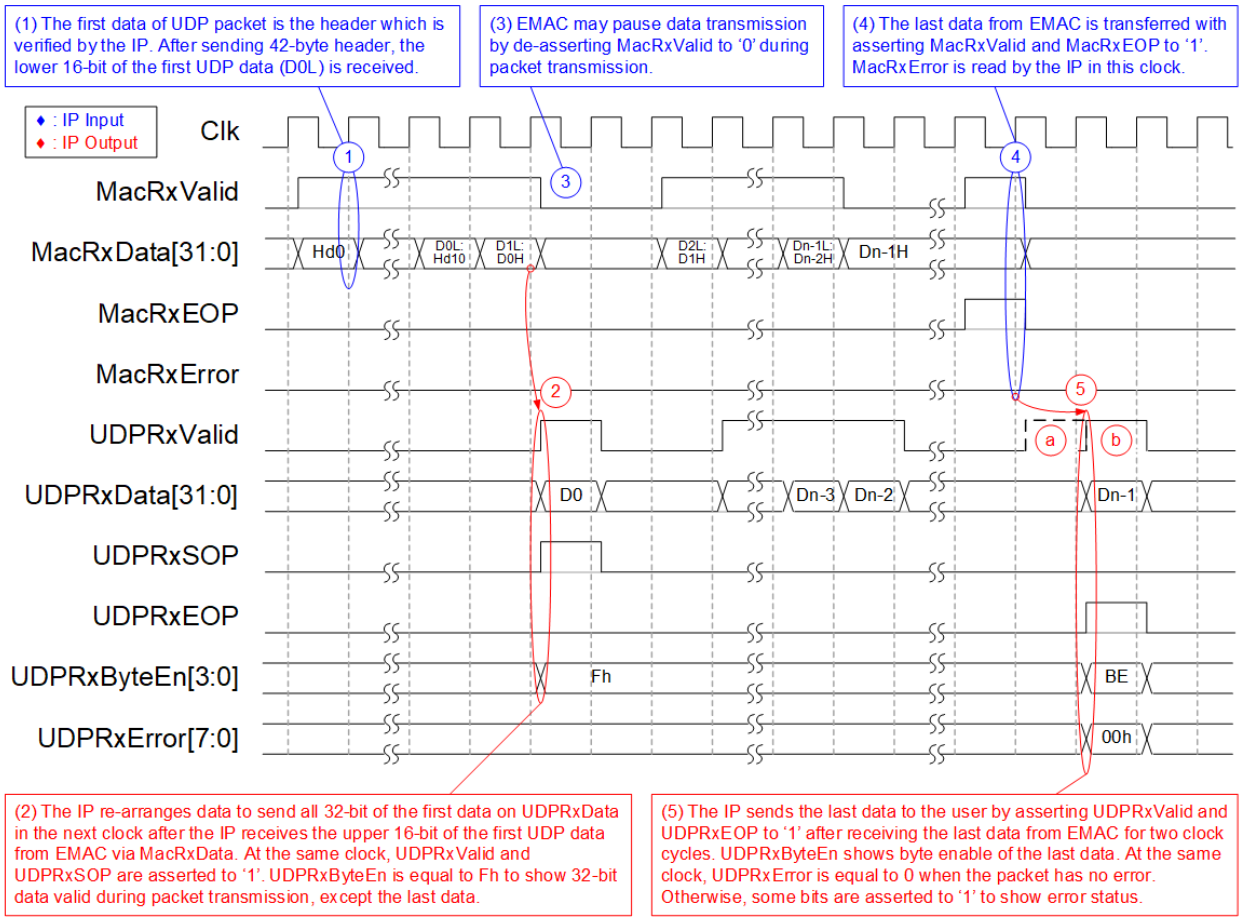

Figure 8: User data interface timing diagram

(1) The first data of UDP packet is the header (Hd0) which is sent to the IP with asserting MacRxValid to '1'. The header size of UDP packet is 42 bytes, so the 11th data of the packet consists of 16-bit header and 16-bit UDP data which is the first UDP data. The IP needs to wait the 12th data of the packet to get the upper 16-bit of the first UDP data for forwarding to the user.

(2) In the next clock, the IP merges all 32-bit of the first data to send to the user. The IP asserts UDPRxValid and UDPRxSOP to '1' for sending the first data. UDPRxByteEn is equal to Fh when sending every data to the user except the last data which may be equal to 0001b (1-byte), 0011b (2-byte), 0111b (3-byte), or 1111b (4-byte).

(3) During the packet transmission, it is possible that MacRxValid from EMAC is de-asserted to '0' to pause data transmission. So, UDPRxValid of the IP is also de-asserted to '0' during waiting the next data from EMAC.

(4) When EMAC sends the last data of the packet, MacRxValid and MacRxEOP are asserted to '1'. In this clock cycle, MacRxError from EMAC is valid to read by the IP to check packet error status.
(5) The IP sends the last data of the packet to the user by asserting UDPRxValid and UDPRxEOP to '1'. If some bytes are valid, UDPRxByteEn is not equal to Fh. Also, UDPRxError shows the error status of the packet in this clock cycle. If some errors are found, UDPRxError is not equal to 00h. As shown in Figure 8, there are two timing diagrams for sending end of packet to the IP.
    a) When the last data from EMAC is valid for 3-4 bytes, the IP must split the last data to send by using two data. So, UDPRxValid is asserted to '1' for two clock cycles. UDPRxByteEn is equal to 0001b or 0011b for sending remaining one and two bytes of the last data respectively.
    b) When the last data from EMAC is valid for 1-2 byte, the last data can be packed with the previous data to sent within one data. So, UDPRxValid is asserted to '1' for one clock cycle. UDPRxByteEn is equal to 0111b or 1111b for sending three or four bytes of the last data respectively.

Session Closing (Multicast mode)

To change the parameters of some active sessions, SSEnable of the session must be de-asserted to '0'. Figure 9 shows the example to disable two sessions, i.e. session#0 and #2 in Multicast mode.

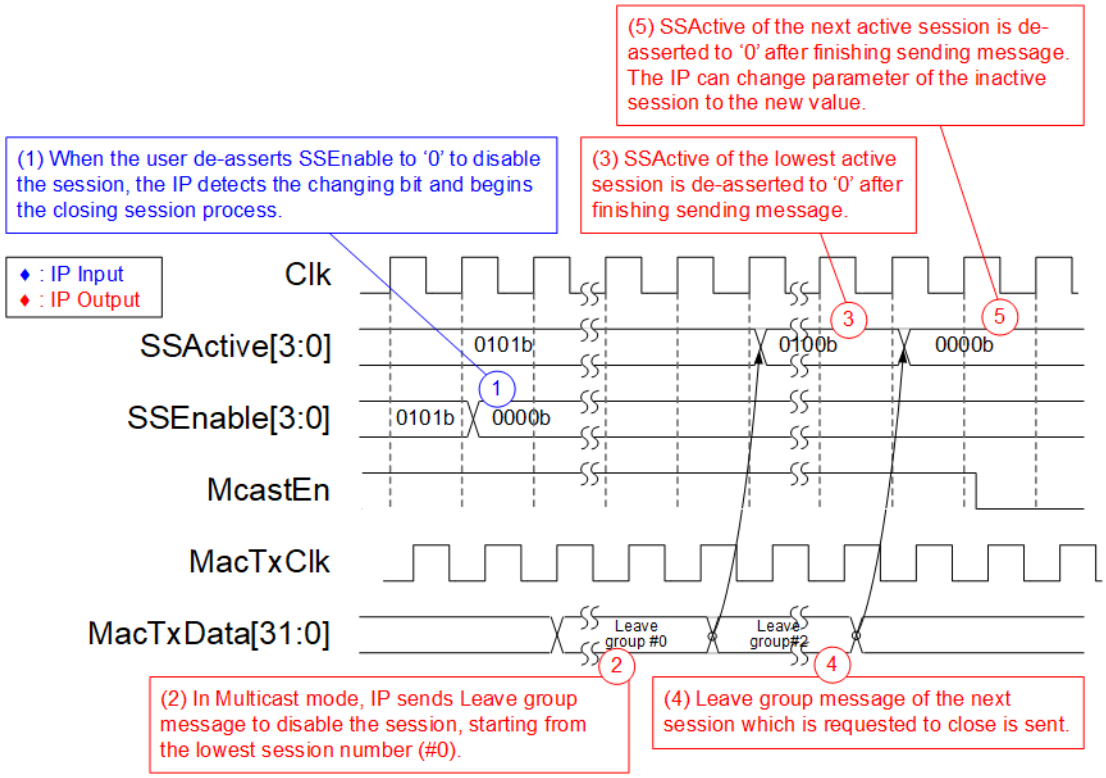

Figure 9: Multicast session closing

(1) Some bits of SSEnable are de-asserted to '0' to disable the session. The IP detects the change and runs the process to close the requested session, starting from the lowest session number. In the example, two sessions are requested, session#0 and #2, so session#0 is firstly closed.
(2) In Multicast mode (McastEn='1'), IP sends leave group message by using session#0 parameters.
(3) After finishing sending the message, SSActive of the completely closed session (session#0) is de-asserted to '0'.
(4) The leave group message is sent to close the next session (session#2) following the user request until SSEnable is equal to SSActive.
(5) After finishing closing all close requested session, the user can change the parameters of the inactive session. When SSEnable is re-asserted to '1', the IP loads the new parameters and begins receiving the data.

Session Closing (Unicast mode)

In Unicast mode, there is no message generated when closing the session but the IP has the internal process to disable the session. Similar to Multicast mode, only one session is closed at a time, starting from the lowest session number, as shown in Figure 10.

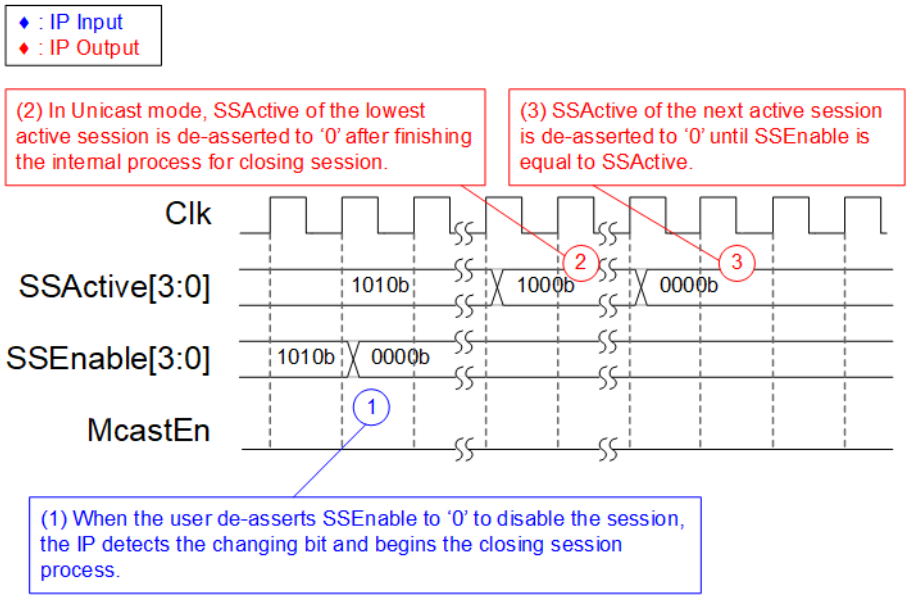

Figure 10: Unicast session closing

(1) Some bits of SSEnable are de-asserted to '0' to disable the session. The IP detects the change and runs the process to close the request session, starting from the lowest session number. In the example, two sessions are requested to close, so session#1 is firstly closed.

(2) In Unicast mode (McastEn='0'), after the IP finishes the internal process to close the session, SSActive of the completely closed session is de-asserted to '0'.

(3) The internal process to close each session is run until SSEnable is equal to SSActive. After that, the user can change the parameters of the inactive session. The IP loads the new paramters when SSEnable of the inactive session changes from '0' to '1'.

Verification Methods

The UDP10GRx 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	Jun-4-2020	New release