

UDP10GRx IP Core

April 21, 2021

Product Specification

Rev1.0



Design Gateway Co.,Ltd

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

URL: www.design-gateway.com

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
- 3.1 ns latency time for receiving data, measured from UDP payload data to user data at 322.266 MHz
- 32-bit Avalon stream to interface with DG LL10GEMAC IP or Intel Low latency Ethernet 10G MAC IP
- Individual clock domain for transmit and receive interface at 322.266 MHz/312.5 MHz
- Reference design available on Arria10 GX development board
- Customized service for following features
 - Additional sessions
 - IGMPv3
 - IP fragmentation

Core Facts	
Provided with Core	
Documentation	Reference design manual, Demo instruction manual
Design File Formats	Encrypted HDL
Instantiation Templates	VHDL
Reference Designs & Application Notes	QuartusII Project, See Reference Design Manual
Additional Items	Demo on Arria10 GX board
Support	
Support provided by Design Gateway Co., Ltd.	

Table 1: Example Implementation Statistics

Family	Example Device	Fmax (MHz)	ALMs ¹	Registers ¹	Pin	Block Memory bit ²	Design Tools
Arria 10 GX	10AX115S2F4511SG	322.266	1045	1410	-	0	QuartusII18.0

Notes:

1) Actual logic resource dependent on percentage of unrelated logic

Applications

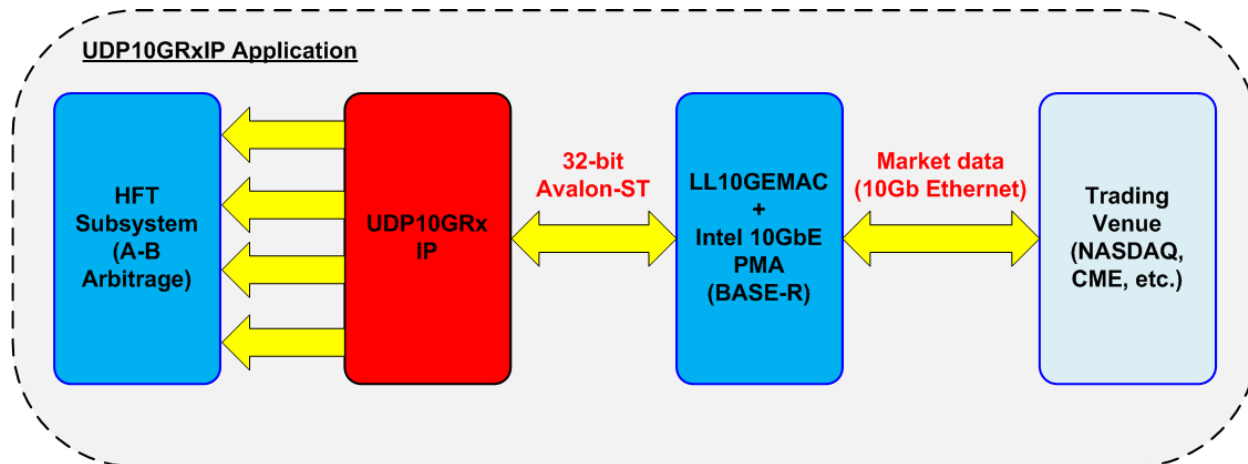


Figure 1: UDP10GRx IP Application

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 provided by DesignGateway is shown in Figure 1.

Reference Design

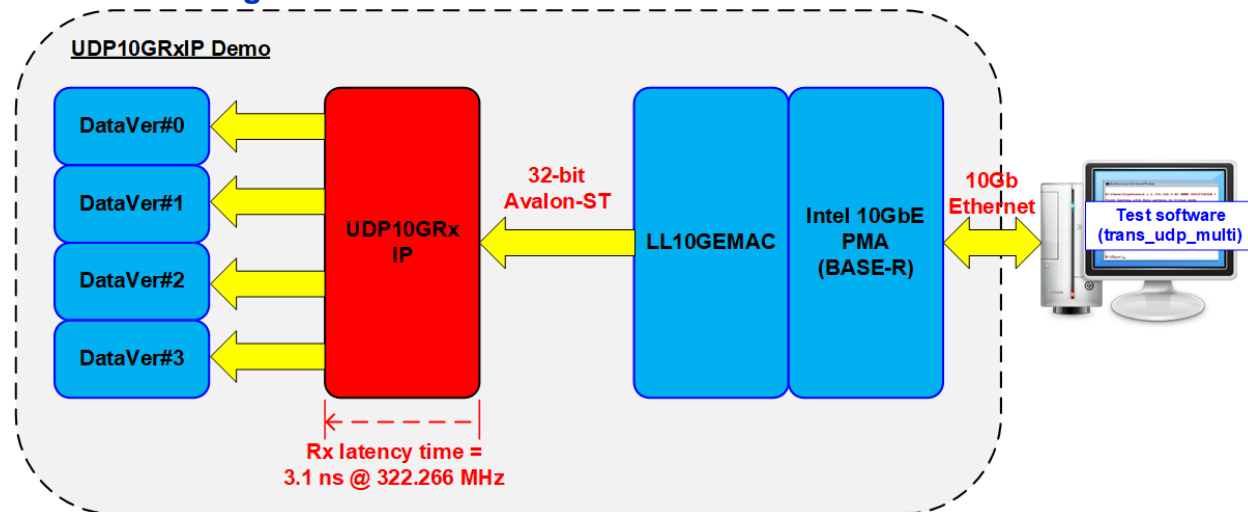


Figure 2: UDP10GRx Latency

UDP10GRx IP has been launched with the standard demo design on Intel development board. In the demo, the IP is connected with LL10GEMAC IP and Intel 10GbE PMA (BASE-R), for receiving UDP data transferred from test application run on Test PC. Up to four sessions can be transferred at the same time. User interface of UDP10GRx IP is connected to the data verification logic. One session of user interface is connected to one verification block. Latency time for receiving data in UDP10GRx IP is 3.1 ns when running by 322.266 MHz, measured from UDP payload data to the first user data output of UDP10GRx IP).

General Description

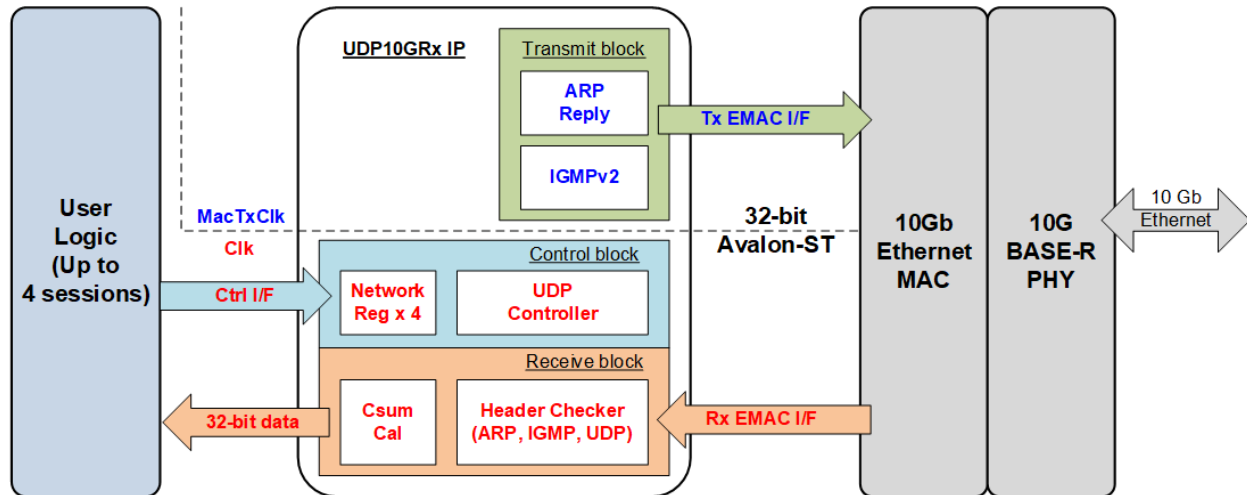


Figure 3: UDP10GRx IP Block Diagram

UDP10GRx IP implements the logic to decode UDP packet stream 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 are supported, 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 - common inputs and session inputs. Common inputs are the shared inputs for all sessions such as operation mode, Unicast or Multicast. Session inputs are the inputs which are independently assigned 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 values (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 the group communication while Unicast mode is 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 joining group by using Multicast IP which is the session parameter. After that, UDP10GRx IP decodes the received UDP packet when Multicast IP is matched. The decoded data is forwarded 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 Membership Query message of the active session is received, Membership report is returned by UDP10GRx IP.

When running in Unicast mode, ARP protocol is enabled. UDP10GRx IP returns ARP reply when receiving ARP request that has the matched IP address with the active session. Similar to Multicast mode, the UDP10GRx IP decodes the received UDP packet and extracts the data to user until the session is disabled.

If some parameters in the packet header are not matched to the parameters of the active session, the packet will be rejected. When UDP checksum is error or EMAC asserts error flag at the end of Ethernet frame, the packet is still transferred to the user interface but error flag is asserted at the end of packet.

Functional Description

As shown in Figure 3, UDP10GRx IP supports different clock domain for Tx interface and Rx interface with 10G EMAC. However, the clock frequency of 10G EMAC for Tx and Rx interface is similar, 322.266 MHz for low-latency solution. The user interface of UDP10GRx IP uses the same clock as Rx interface of 10G EMAC, named Clk. Another clock, MacTxClk, is applied to be Tx interface of 10G EMAC only.

Control block

This block is interface with the user for controlling the IP operation. Therefore, the parameters from the user are processed in this block.

- **Network Reg**

The registers to store network parameters are divided to five parts, i.e., one common parameter and four session parameters. The common parameters are loaded when the IP changes from inactive to active status to operate some sessions. The session parameters are loaded independently when that session changes from inactive to active status, controlled by enable flag. The parameters do not change if the session is still active.

- **UDP Controller**

The controller can run in two modes - Unicast mode and Multicast mode. When running Unicast mode, UDP controller enables ARP reply submodule inside Transmit block. Otherwise, IGMPv2 submodule is enabled for Multicast mode. Furthermore, UDP controller loads the parameters to Header checker for the packet verification. The controller is run in two phases – the session initialization and the data reception.

Transmit block

This block is interface with the control block to create the packet to 10GEMAC via Tx EMAC interface. Therefore, this module is run on MacTxClk, the same clock as Tx EMAC interface.

- **ARP Reply**

This module is enabled when running in Unicast mode. ARP reply packet is created to be the response packet of the valid ARP request.

- **IGMPv2**

This module is enabled when running in Multicast mode. The message is created for joining group or leaving group following IGMPv2 protocol.

Receive block

This block is interface with the user for 32-bit received data, extracted from the received packet of Rx EMAC I/F. The packet is validated and extracted in this block.

- **Header Checker**

The checker supports three packet types – ARP packet, IGMP packet, and UDP packet. There are four session parameters for comparing the received packet. However, only the parameters of the active session is applied for the packet verification. ARP packet is valid in Unicast mode while Membership Query (IGMPv2) is valid in Multicast mode. The UDP data of the valid UDP packet is extracted and forwarded to 32-bit data user interface. If EMAC asserts error flag at the end of packet, the IP also forwards the error to the user for cancelling the packet at the end of packet.

- **Csum Cal**

This module is designed to calculate IP checksum and UDP checksum of the received packet. If UDP checksum is not correct while the other header is valid, the received data of the packet is forwarded to the customer but the error flag is asserted at the end of packet, similar to the condition that EMAC asserts the 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_A_E.html

Another solution of 10Gb Ethernet MAC is Low Latency Ethernet 10G MAC, provided by Intel. 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.intel.com/content/www/us/en/programmable/products/intellectual-property/ip/interface-protocols/m-alt-10gbps-ethernet-mac.html>

User Logic

The user interface of UDP10GRx IP is divided to two groups, i.e., control interface and data interface. Control interface consists of control signals and the network parameter settings. The user can assign the network parameters by using registers or constant value. The data interface is 32-bit interface by using general stream interface, controlled by valid signal. There are four bits of valid signal for four session controlling while the data bus are shared for all sessions.

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 Rx EMAC interface. When running with DG LL10GEMAC, the clock frequency is equal to 322.266 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. <i>Note: The IP loads McastEn, SrcMacAddr, and SrcIPAddr when SSEnable changes from 0000b to other values (some sessions are active).</i>
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. One bit is applied for one session. '0'-Disable, '1'-Enable. Bit[0], [1], [2], and [3] - Session#0, #1, #2, and #3 respectively. <i>Note: Before asserting SSEnable to '1', SSActive of that session must be equal to '0' (Inactive).</i>
SSActive[3:0]	Out	Session status returned by the IP. One bit is the status of one session. '0'-Inactive, '1'-Active. Bit[0], [1], [2], and [3] - Session#0, #1, #2, and #3 respectively.
McastIPAddr0-3[31:0]	In	Multicast IP address for each session. Use only when the IP runs in Multicast mode. <i>Note:</i> (1) The IP loads the parameters for each session when SSEnable of that session changes from '0' to '1'. (2) The number after the parameter name refers to the session number that is equal to 0 – 3 for Session#0 - #3. 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 four sessions, one bit per session. Only one bit can be asserted to '1' at a time. Bit[0], [1], [2], and [3] are the data valid of Session#0, #1, #2, and #3 respectively.
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 1111b (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.
MacTxEmpty[1:0]	Out	Specify the number of bytes which are unused of the last word in the frame. This signal is equal to 00b for enabling all 32-bit data in a packet except the last data of the packet. 00b: 4-byte valid, 01b: MacTxData[23:0] valid, 10b: MacTxData[15:0] valid, 11b: MacTxData[7:0] valid. The signal is valid when MacTxValid='1'.
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, MacTxEmpty, MacTxSOP, and MacTxEOP) must hold the same value until MacTxReady is re-asserted to '1'.

Note: According to Intel 10G Ethernet MAC specification, the data endian of UDPRxData and MacRxData are swapped for Rx interface. While the first data byte of UDPRxData is transferred by bit[7:0], the first byte of MacRxData is transferred by bit[31:24].

Timing Diagram

IP Reset

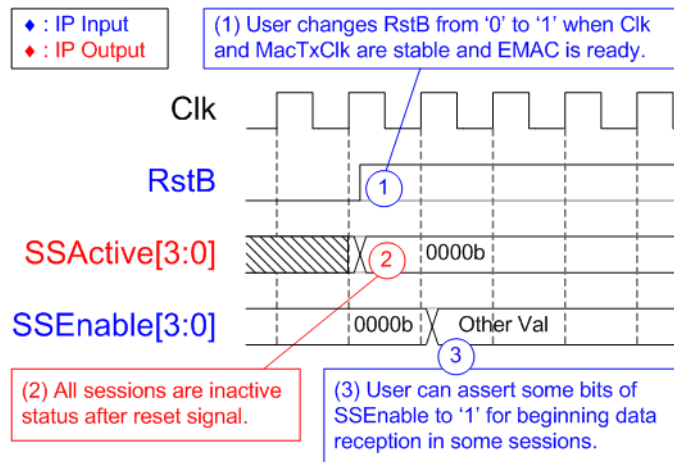


Figure 4: IP Reset process

- (1) RstB is de-asserted to '1' by user logic after both clock inputs, Clk and MacTxClk, to UDP10GRx IP are stable. Also, EMAC should be ready for transferring data with the IP.
- (2) SSetActive, output from UDP10GRx IP, is equal to 0h after reset process to show all sessions are inactive. Now the IP is ready to operate.
- (3) 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 shared for all sessions and session parameters which are independently defined for each session. Timing diagram for loading common parameters are shown in Figure 5.

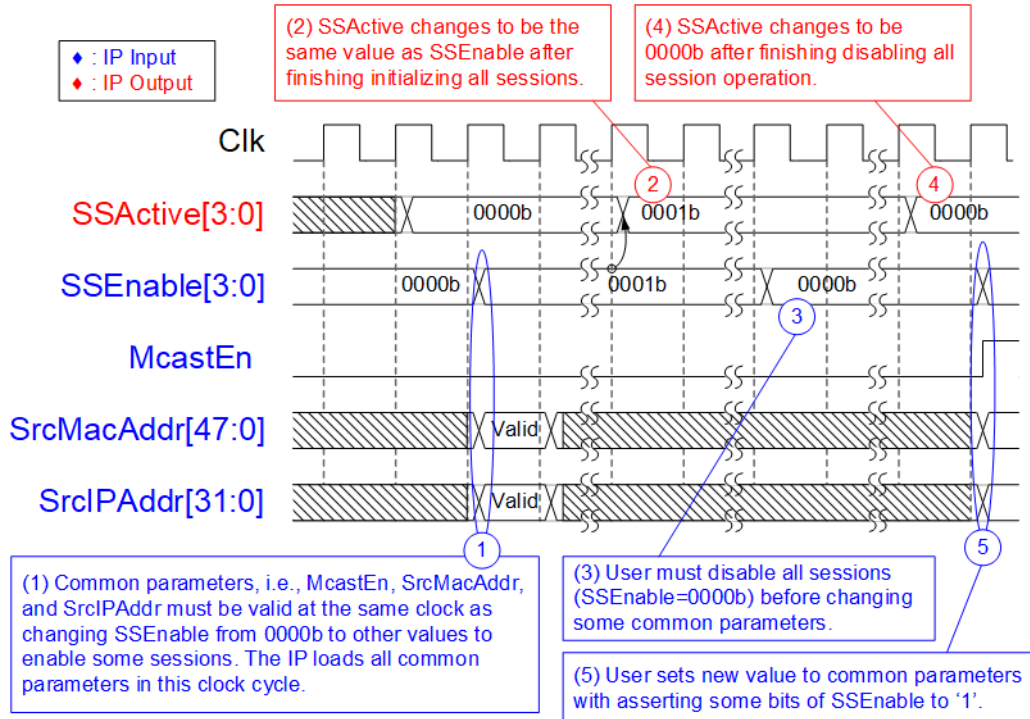


Figure 5: Initialization for common parameters

- (1) When user begins the IP operations by changing SSEnable from 0000b to other values (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 is equal to SSEnable. Now the IP is ready to receive the packet which has the parameters matching to the active session. UDP data of the received valid packet is extracted and forwarded to the user.
- (3) When the user finishes all 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 values to common parameters and assert some bits of SSEnable to '1' for beginning the IP initialization by using the new common 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.

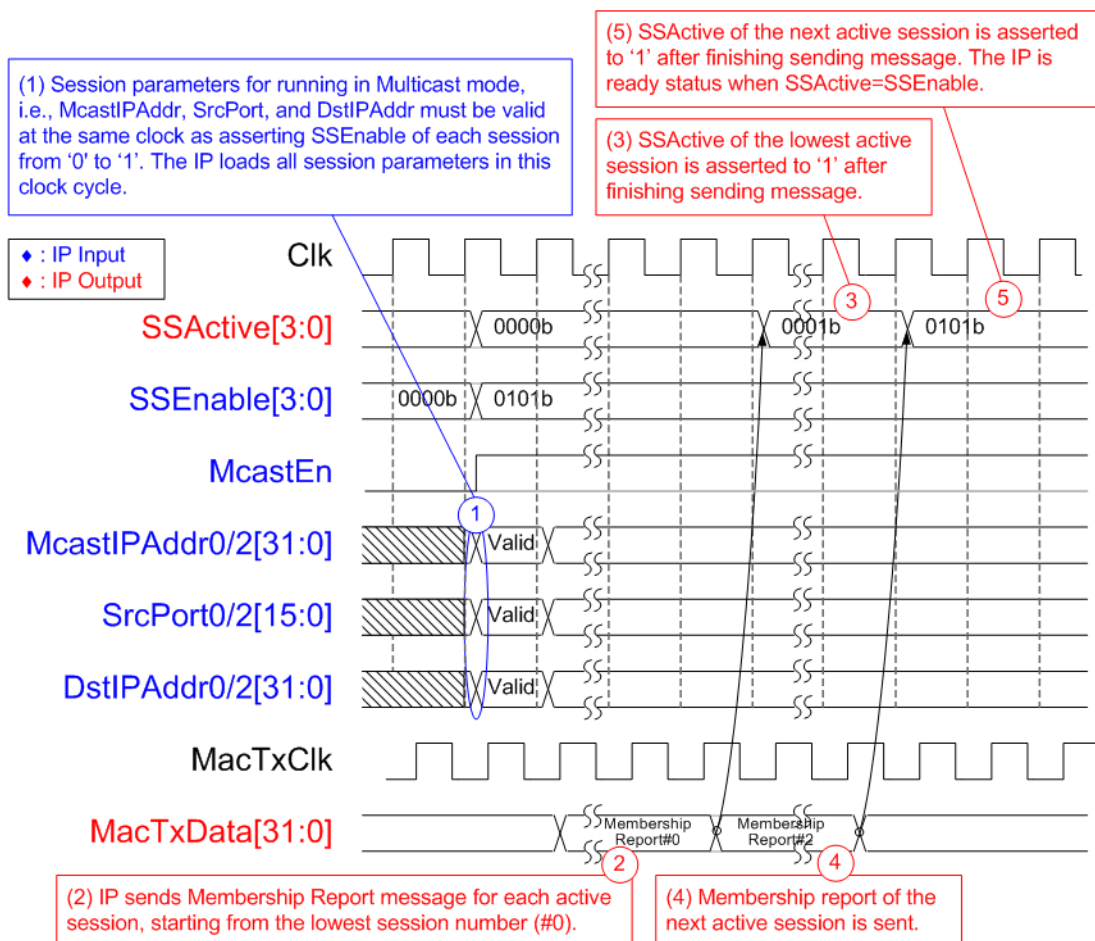


Figure 6: Multicast initialization

- (1) When some bits of SSetActive change from '0' to '1', the session parameters of the enabled session are loaded to the IP in the same clock. In Multicast mode, there are three session parameters, 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, SSetActive 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) SSetActive of the next session (session#2) is asserted to '1'. Repeat step 4) – 5) until SSetActive 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.

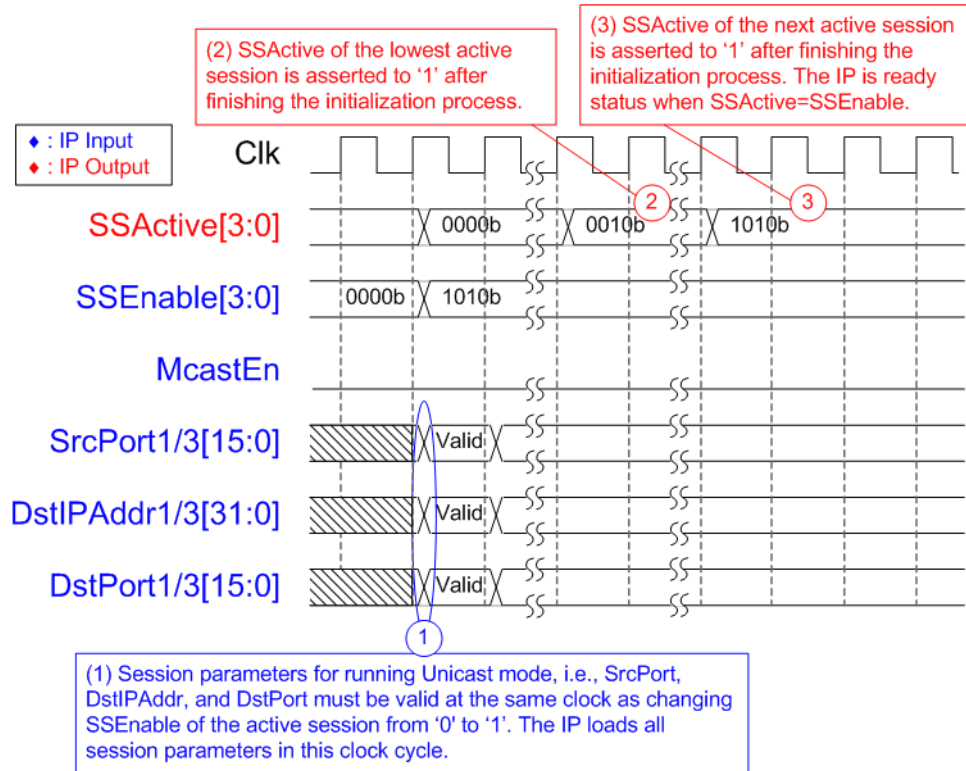


Figure 7: Unicast initialization

- (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, there are three session parameters, 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 SSActive 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 UDP checksum of a packet is not correct.

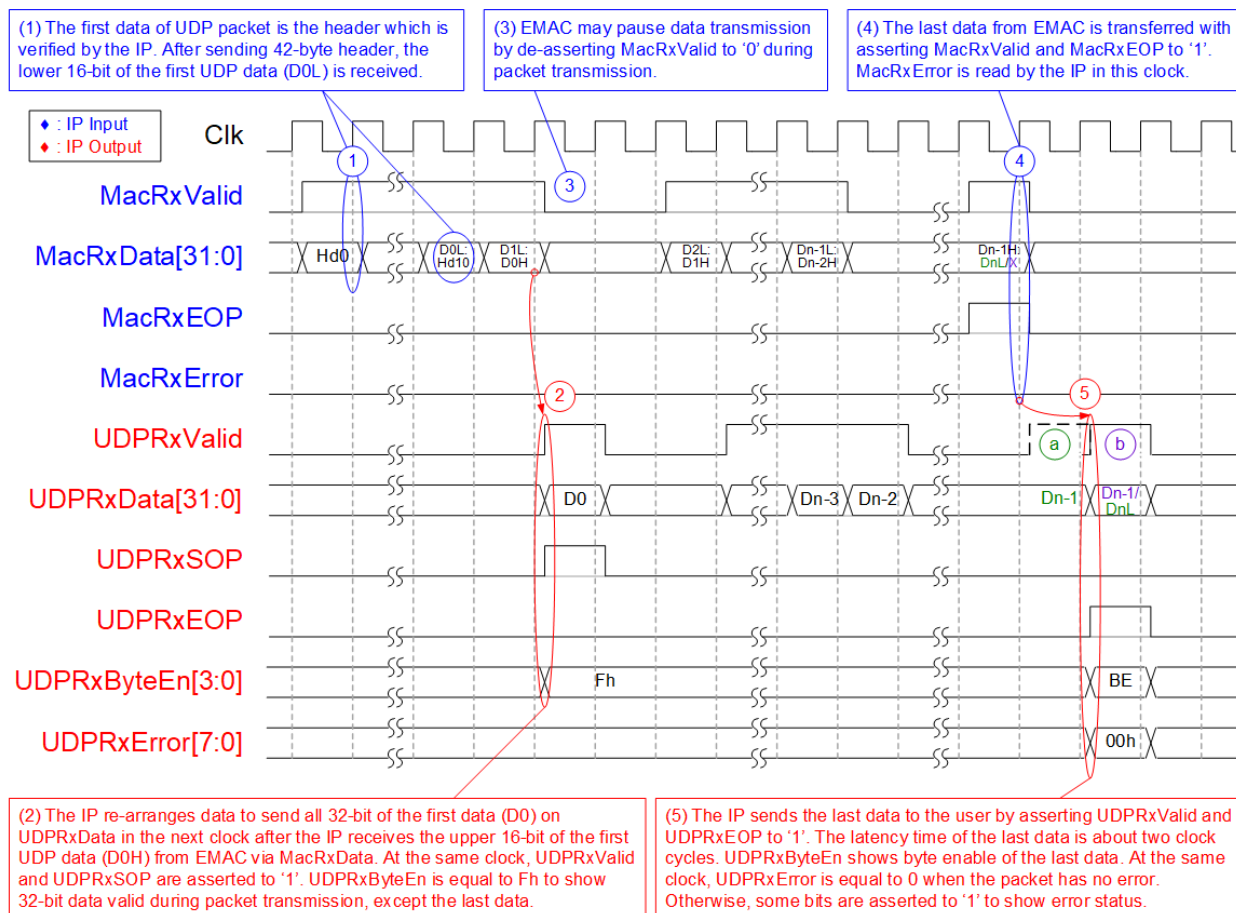


Figure 8: User data interface timing diagram

- (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 (D0L) which is the first UDP data. The IP waits until the 12th data of the packet is received to get the upper 16-bit of the first UDP data.
- (2) In the next clock, the IP merges all 32-bit of the first data (D0) 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).

Note:

- (a) As shown in Figure 8, if MacRxValid is asserted continuously, the minimum latency time measured from the first payload data of EMAC (D0H on MacRxData) to the first data on user interface (D0 on UDPRxData) is equal to 1 clock cycles or 3.1 ns @ 322.266 MHz.
- (b) To measure latency time from start-of-packet of EMAC (MacRxSOP='1') to start-of-packet of user interface (UDPRxSOP='1'), the minimum latency time must include the header data of each UDP packet which transfers at least 11 cycles. Therefore, the minimum latency time measured by start-of-frame is equal to 12 cycles or 37.2 ns @ 322.266 MHz.
- (3) During the packet transmission, it is possible that MacRxValid from EMAC is de-asserted to '0' to pause data transmission. Therefore, UDPRxValid of the IP is also de-asserted to '0' during waiting for the next data from EMAC.
- (4) When EMAC sends the last data of the packet, MacRxValid and MacRxEOP are asserted to '1'. The IP reads MacRxError from EMAC in this cycle to confirm the packet is valid. If EMAC error is detected, the IP asserts UDPRxError[1] to '1'.
- (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. 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 (DnL is valid), UDPRxValid is asserted to '1' for two clock cycles continuously to send Dn-1 and DnL data. UDPRxByteEn of the last data is equal to 0001b or 0011b for sending remaining one-byte or two-byte data respectively.
- (b) When the last data from EMAC is valid for 1-2 byte (Dn-1H is valid without DnL), UDPRxValid is asserted for one cycle to send Dn-1 data. UDPRxByteEn of the last data is equal to 0111b or 1111b for sending three-byte or four-byte 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.

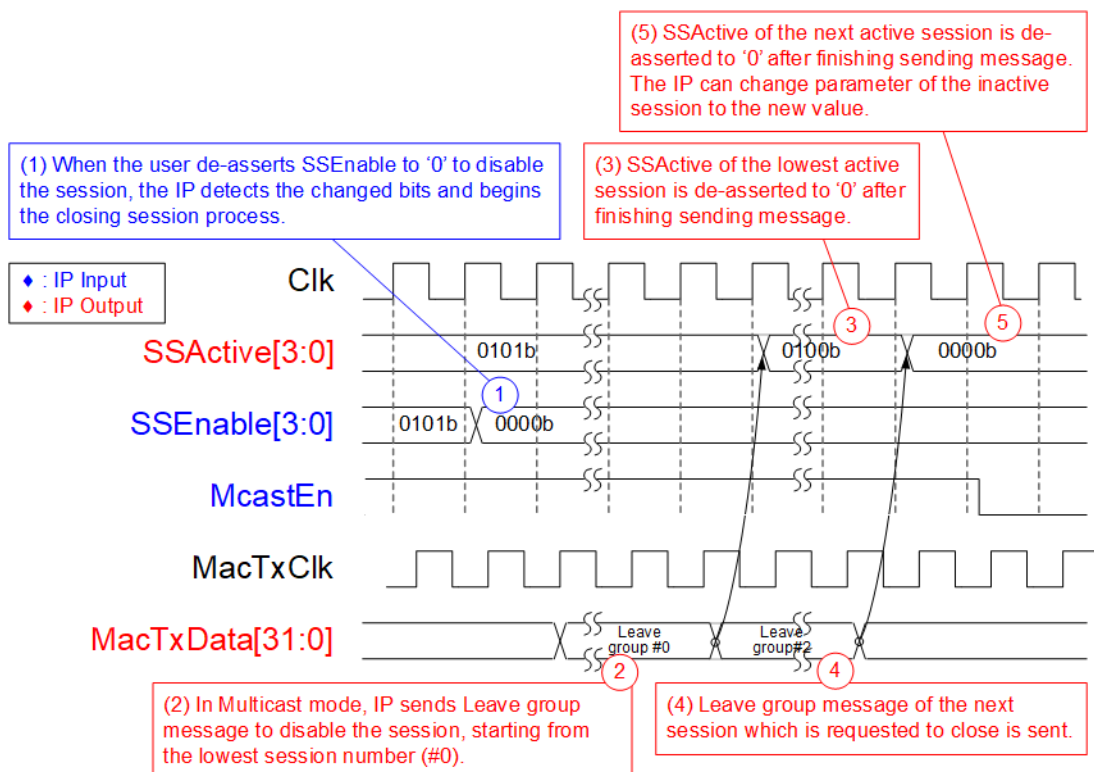


Figure 9: Multicast session closing

- (1) Some bits of SSEnable are de-asserted to '0' to disable the session. The IP detects the changed bits 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 session#2. Therefore, session#0 is firstly closed.
- (2) In Multicast mode (McastEn='1'), the session is disabled by sending leave group message. The leave group by session#0 parameters is created.
- (3) After finishing sending the message, SSetActive of the completely closed session (session#0) is de-asserted to '0'.
- (4) The leave group message by using session#2 parameters is sent to close session#2 which is the next session requested by user. The close operation is finished when SSEnable is equal to SSetActive.
- (5) After finishing closing operation, the user can change the parameters of the inactive session. Next, SSEnable is re-asserted to '1' by user and the IP loads the new parameters for receiving the data.

Session Closing (Unicast mode)

In Unicast mode, there is no message generated when closing the session. After closing the session, ARP process of the inactive session is not operated. Similar to Multicast mode, only one session is closed at a time, starting from the lowest session number, as shown in Figure 10.

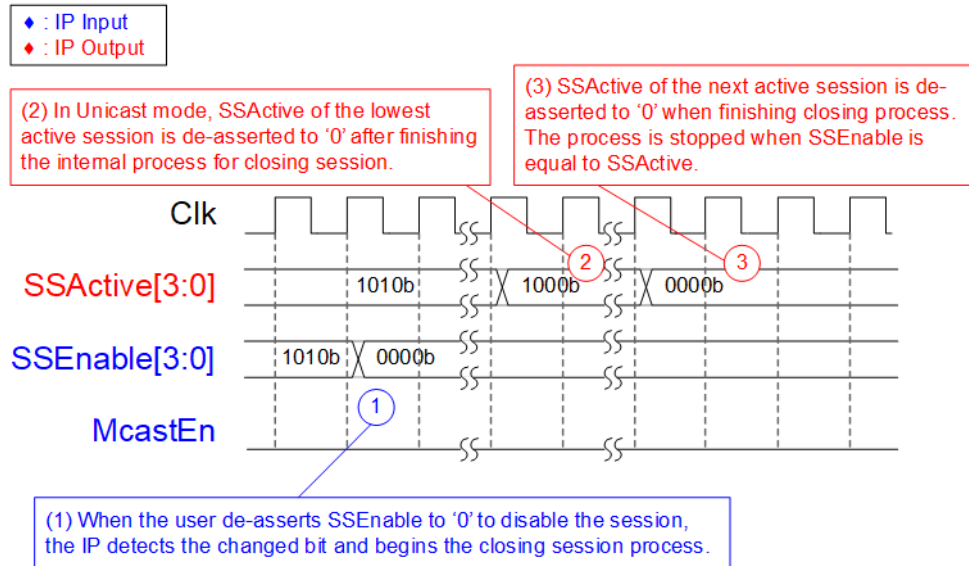


Figure 10: Unicast session closing

- (1) Some bits of SSEnable are de-asserted to '0' to disable the session. The IP detects the changed bits and runs the process to close the requested session, starting from the lowest session number. In the example, two sessions are requested to close, session#1 and session#3. Therefore, session#1 is firstly closed.
- (2) In Unicast mode (McastEn='0'), the IP clears the internal process to close the session. After finishing, SSetActive of the completely closed session is de-asserted to '0'. In this example, bit[1] is de-asserted and then bit[3] is deasserted to complete close operation.
- (3) The close operation is finished when SSEnable is equal to SSetActive. After that, the user can change the parameters of the inactive session. The IP loads the new paramters when SSEnable of the 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 Arria10 GX development 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 Gateway Co., Ltd. for pricing and additional information about this product using the contact information on the front page of this datasheet.

Revision History

Revision	Date	Description
1.0	21-Apr-21	New release