

UDP10GRx IP Core

November 22, 2021

Product Specification

Rev2.0



Design Gateway Co.,Ltd

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

URL: 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
- Support ICMP echo reply (Ping)
- Operation mode: Unicast or Multicast (IGMPv3)
- Configuration mode: Master or Slave (for multiple IP connecting)
- Up to four sessions per one UDP10GRx IP
- 37.2 – 40.3 ns latency time for receiving data (12-13 cycles of 322.266 MHz), measured from start-of-packet to start-of-packet
- 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 (322.266 MHz or 312.5 MHz)
- Available reference design
 - 4 and 16 sessions demo on Xilinx development board (ZCU102/KCU116)
 - Accelerated Algorithm Trading (AAT) demo on Xilinx Accelerator cards (U50/U250)
- Customized service for following features
 - 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	Vivado Project, See Reference Design Manual
Additional Items	Demo on ZCU102/KCU116
Support	
Support Provided by Design Gateway Co., Ltd.	

Table 1: Example Implementation Statistics

Family	Example Device	Mode	Fmax (MHz)	CLB Regs	CLB LUTs	CLB	IOB	BRAMTile	Design Tools
Kintex-Ultrascale	XCKU040FFVA1156-2E	Master	322.266	1619	1640	361	-	-	Vivado2019.1
		Slave		1168	1370	302	-	-	
Kintex-Ultrascale+	XCKU5PFFVB676-2E	Master	322.266	1619	1615	377	-	-	Vivado2019.1
		Slave		1168	1354	332	-	-	
Zynq-Ultrascale+	XCZU9EGFFVB1156-2I	Master	322.266	1619	1615	382	-	-	Vivado2019.1
		Slave		1168	1350	311	-	-	
Virtex-Ultrascale+	XCVCU9PFLGA2104-2L	Master	322.266	1619	1616	390	-	-	Vivado2019.1
		Slave		1168	1350	316	-	-	

Applications

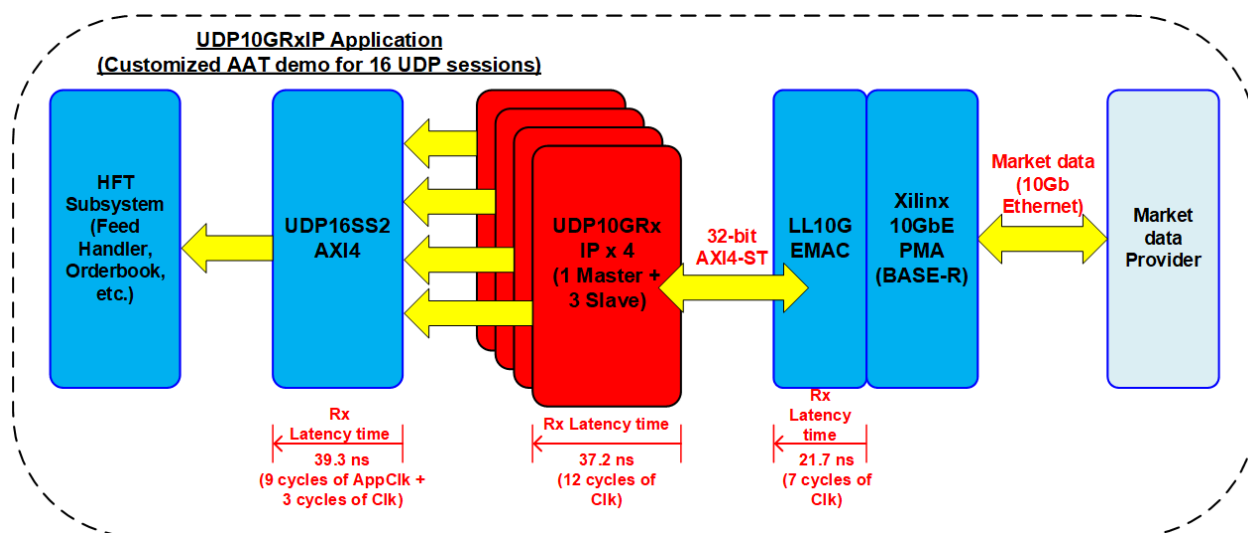


Figure 1: UDP10GRx IP Application

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 extracts UDP payload data from Ethernet frame via 10Gb Ethernet (BASE-R). Up to four sessions can be decoded by using one UDP10GRx IP.

Figure 1 shows the example application of UDP10GRx IP by integrating the IP to Accelerated Algorithm Trading demo for processing sample market data stream that is transmitted by using UDP/IP protocol. Two DG low latency IPs are integrated, one LL10GEMAC IP and four UDP10GRx IPs. Four UDP10GRx IPs are applied to support 16 session data processing, one configured by Master and three configured by Slave. UDP16SS2AXI4 is designed to transfer 16 session UDP payload data from four UDP10GRx IPs to one AXI4-ST interface. HFT Subsystem includes many blocks for processing market data such as Feed Handler and Orderbook which are coded by HLS. Using DG low latency solution extracts UDP payload data from Ethernet frame with very low latency time. To receive Ethernet packet from Xilinx 10Gb PMA (BASE-R), LL10GEMAC IP and UDP10GRx IP are run at clock output from PMA which is equal to 322.266 MHz. Therefore, latency time of both IPs in Figure 1 is measured by this clock. While UDP16SS2AXI4 module is the adapter logic that includes asynchronous logic. Therefore, latency time of UDP16SS2AXI4 is measured by two clocks, Clock of EMAC (Clk) at 322.266 MHz and Application clock (AppClk) at 300 MHz.

Default AAT demo that is provided by Xilinx can be downloaded from following link.

<https://www.xilinx.com/applications/data-center/financial-technology/accelerated-algorithmic-trading.html>

General Description

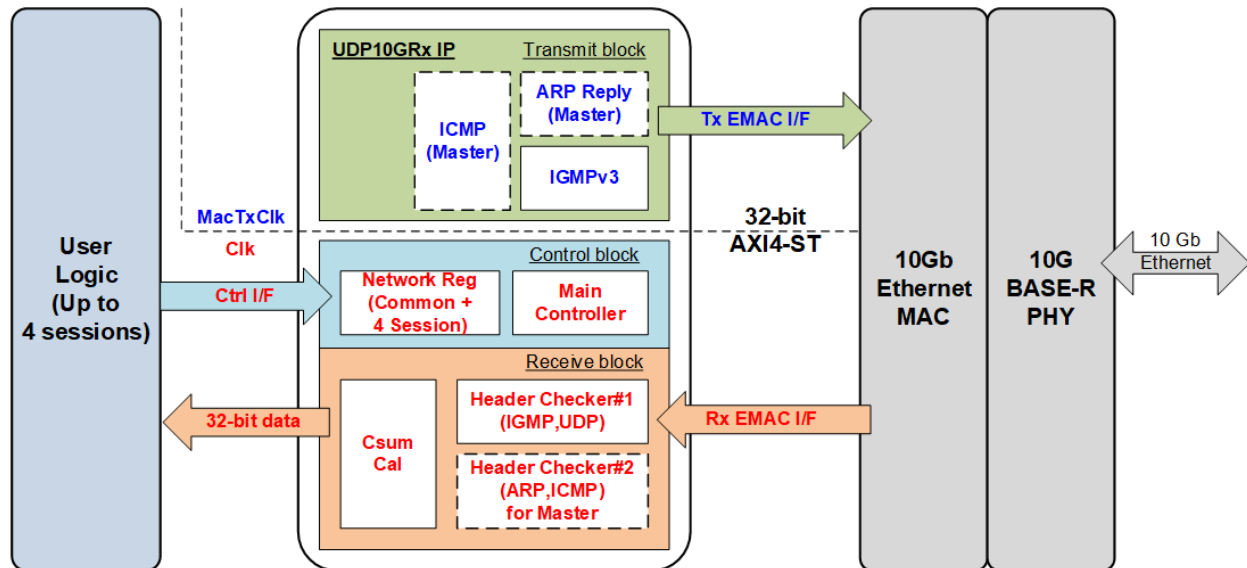


Figure 2: UDP10GRx IP Block diagram

UDP10GRx IP implements the logic to decode UDP packet stream in Multicast and Unicast mode. Up to four network parameters for receiving data of four sessions can be set by user. Some parameters of four sessions are shared such as the IP address and MAC address of UDP10GRx IP, called common parameters. While most parameters are independently set for each session such as IP address of the target.

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 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'. UDP10GRx IP has two operation modes. 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 two Membership reports with 1.2 ms gap time to join group by using Multicast IP. After that, UDP10GRx IP can accept the UDP packet, broadcasted to Multicast IP. 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 two times with 1.2 ms gap time by UDP10GRx IP. If General or Group specific membership query 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 is requested to IP address of UDP10GRx IP. Similar to Multicast mode, The UDP10GRx IP extracts the UDP payload data to user when the header packet is correct until the session is disabled.

Besides, UDP10GRx IP supports Ping which is applied to measure round-trip time. UDP10GRx IP returns ICMP Echo reply after receiving ICMP Echo request.

If some parameters in the packet header do not match 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.

To increase the number of session for processing, multiple UDP10GRx IPs must be applied. Typically, multiple IPs are connected to the same EMAC and shares the same common parameters. When receiving ARP request or ICMP Echo request, ARP reply or ICMP Echo reply is returned by UDP10GRx IPs that is configured by Master mode. While UDP10GRx IP configured by Slave mode ignores ARP and ICMP packet. Master mode or Slave mode is assigned by parameters in HDL code. When using multiple IPs, it is recommended to configure one IP to be Master mode and other IPs to be Slave mode to return one ARP reply/ICMP Echo reply when receiving one ARP request/ICMP Echo request.

Functional Description

When 10G EMAC is configured to low latency mode, the Tx interface and Rx interface of EMAC are run in different clock domain but the same frequency, 322.266 MHz. Therefore, UDP10GRx IP supports to connect with Tx and Rx interface of EMAC by using different clock domain, MacTxClk (Tx I/F) and Clk (Rx I/F). The user interface of UDP10GRx IP is synchronous with Clk. As shown in Figure 2, the logic of UDP10GRx IP consists of three parts, i.e., Control block to configure network parameters, Transmit block to create packet, and Receive block to decode packet.

Control block

This block is user interface for controlling the IP operation and setting the parameters. Therefore, the parameters are pre-processed in this block. Besides, it includes the controller to control the sequence of transmitted packet and received packet which depends on each packet type.

- **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 the session changes from inactive to active status, controlled by session enable flag. The parameters cannot change if the session is still active

- **Main Controller**

The controller is run in two phases – the session initialization and the data transmission. After the parameters are loaded to Network Reg, the latch parameters are loaded to Transmit block for creating the packet header of transmitted packet and Receive block for verifying the packet header of received packet. After that, the controller changes to data transmission phase.

In data transmission phase, the controller can be set to two operation modes - Unicast mode and Multicast mode. When running Unicast mode, the controller enables ARP reply submodule inside Transmit block and loads Unicast parameters to Receive block. Otherwise, IGMPv3 submodule is enabled for Multicast mode and Receive block loads Multicast parameters. The controller sets the created packet type and enables Transmit block to generate the packet after decoding the packet type of received packet, returned by Receive block.

Transmit block

This block is the interface module with Tx EMAC interface to create the packet. Therefore, this module is run on MacTxClk, the same clock as Tx EMAC interface. There are three packet types created by this block, i.e., ICMP Echo reply, ARP reply, and IGMP. ARP Reply and ICMP are included when IP is configured by Master mode only. ARP reply is run when the session is enabled by Unicast mode. IGMPv3 is run when the session is enabled by Multicast mode.

- **ARP Reply**

ARP reply packet is returned by this module after receiving ARP request. Header data of ARP reply packet consists of the parameters that are extracted from ARP request packet and the common parameters that are loaded from Control block.

- **IGMPv3**

This module is always integrated for both Master and Slave configuration mode. It is run when the session is enabled by Multicast mode. To support IGMPv3 protocol, two message types must be created, Membership report and Leave group message. The message type and number of message are set by Control block. The header data consists of the common parameters and the session parameters that are loaded from Control block.

- **ICMP**

This module creates ICMP Echo reply packet after receiving ICMP Echo request. ICMP Echo reply packet consists of the header and the data. Similar to ARP reply, ICMP header loads some parameters from ICMP Echo request and the command parameters from Control block. ICMP data must be extracted from ICMP Echo request to store to small buffer. After that, ICMP module reads ICMP data from the small buffer to create ICMP Echo reply packet. The buffer size is not large to optimize the resource utilization, so the maximum ICMP data size is 118 bytes. Please contact our sales if larger ICMP data size is required.

Receive Block

This block decodes the received packet from EMAC. If UDP payload data of the active session is received, the data will be extracted and forwarded to user via 32-bit data interface. Received block supports four packet types, i.e., IGMP, UDP, ARP, and ICMP. ARP and ICMP packet type are supported when UDP10GRx IP is configured in Master mode only. ARP is run when the session is enabled by Unicast mode. IGMP is run when the session is enabled by Multicast mode.

- **Header Checker#1 and #2**

There are two Header checkers inside Receive block. First is designed to verify IGMP and UDP header which is integrated for both Master and Slave mode. Second is designed to verify ARP and ICMP header which is integrated for Master mode only. The parameters to verify the header are loaded from Control block which consists of common parameters and session parameters. Only the parameters of the active session is applied for the packet verification.

The UDP data is extracted and forwarded to user when the network parameters of UDP header are equal to common parameters and session parameters of active session. Also, IP checksum must be correct. There are two possible conditions that UDP data of error packet is sent to user with asserting error flag at the end of packet. First, UDP checksum of the packet is not correct. Second, EMAC asserts error status at the end of received packet. Therefore, user logic needs to ignore the received data when the error is asserted by UDP10GRx IP.

- **Csum Cal**

This module is designed to calculate checksum of the received packet. UDP, ICMP, and IGMP have two checksums that must be calculated, IP checksum and its own protocol. The result is applied to verify with the value in the header of received packet. The packet will be rejected if the checksum value is not correct.

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 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., control interface and data interface. Control interface consists of control signals and the network parameters. The user assigns the network parameters by connecting to registers or constant value. The data interface is 32-bit data stream interface, controlled by valid signal. Also, the session number is sent with the user data to show the session of this data packet.

Core I/O Signals

Descriptions of IP parameters and I/O signals are provided in Table 2 and Table 3.

Table 2: Core Parameters

Parameter	Value	Description
Mode	0, 1	Configuratin mode. '0'-Slave mode, '1'-Master mode In the system, at least one Master must be applied to return ARP and ICMP. Slave mode is applied to optimize resource when multiple UDP10GRx IPs are applied in the system.

Table 3: 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, this clock frequency is equal to 322.266 MHz.
User control interface (Synchronous to Clk)		
Common Parameters		
McastEn	In	IP mode. '0'-Unicast mode, '1'-Multicast mode. All sessions operates 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.
Session Parameters		
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:</i> 1. Before asserting SSEnable to '1', SSActive of that session must be equal to '0' (Inactive). 2. Before updating SSEnable, the value of SSActive must be equal to SSEnable value to confirm that there is no session initializing. If the value is not equal, it needs to wait until SSActive equal to SSEnable to complete the session initialization. 3. The IP loads the session parameters when SSEnable of that session changes from '0' to '1'. 4. The number after the parameter name refers to the session number. For example, Session#0: Use McastIPAddr0, SrcPort0, DstIPAddr0, and DstPort0. Session#1: Use McastIPAddr1, SrcPort1, DstIPAddr1, and DstPort1.
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.
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)		
UDPRxSSNo[1:0]	Out	Session number of UDPRxData. Valid when UDPRxValid='1'. "00"-Session#0, "01"-Session#1, "10"-Session#2, "11"-Session#3.
UDPRxValid	Out	Asserted to '1' when UDPRxData is valid.
UDPRxData[31:0]	Out	UDP data output. Valid when UDPRxValid='1'.
UDPRxByteEn[3:0]	Out	Byte enable of UDPRxData. Valid when UDPRxValid='1'. During packet transmission, this signal is equal to 1111b for enabling all 32-bit data except the final 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. Valid when UDPRxValid='1'.
UDPRxEOP	Out	Asserted to '1' when sending the final data of the packet. Valid when UDPRxValid='1'.
UDPRxError[7:0]	Out	Some bits are asserted to '1' when the received packet has error. Valid when UDPRxValid='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 final 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 final 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 final data of the packet.
MacTxReady	In	Asserted to '1' by EMAC IP when the transmitted data is received successfully. If MacTxReady='0', transmitted data and control signals (MacTxData, MacTxValid, MacTxByteEn, MacTxSOP, and MacTxEOP) must hold the value until MacTxReady is re-asserted to '1'.

Timing Diagram

IP Reset

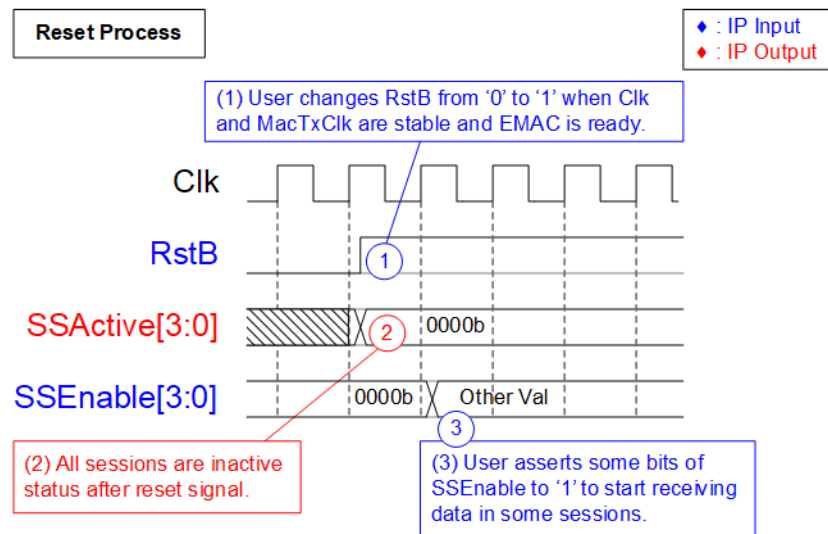


Figure 3: IP Reset process

- (1) RstB is de-asserted to '1' by user logic after both clock inputs of UDP10GRx IP, Clk and MacTxClk, are stable. Also, EMAC must be ready for receiving data transmitted by the IP.
- (2) SSetActive, output from UDP10GRx IP, is equal to 0h after reset process to show all sessions are inactive. The IP is ready to receive parameters and control signals set by user.
- (3) User asserts some bits of SSEnable to '1' to start receiving the data of the selected session from EMAC.

IP Initialization

The IP parameters, input by user, 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 to load common parameters is shown in Figure 4.

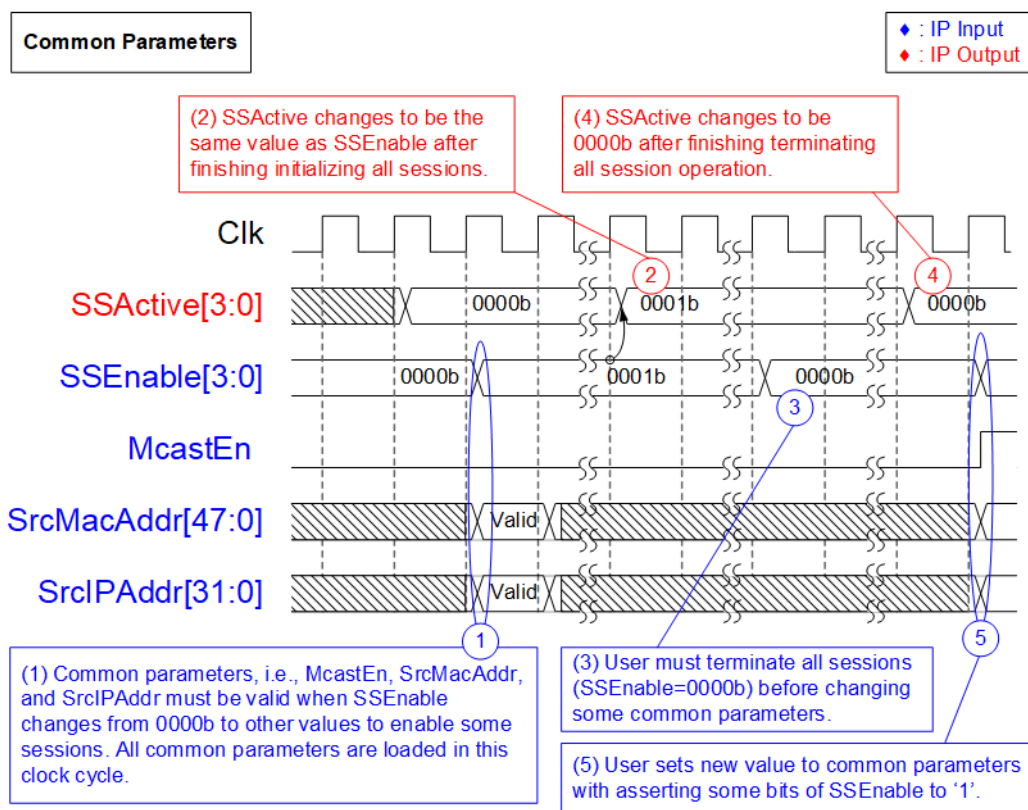


Figure 4: Initialization for common parameters

- (1) The IP loads all common parameters (McastEn, SrcMacAddr, and SrcIPAddr) when SSEnable changes from 0000b (all sessions are disabled) to other values (at least one session is enabled). Therefore, the user must prepare valid value of all common parameters and active session parameters when asserting some bits of SSEnable to '1'. After that, the IP starts session initialization by using the common parameters and the session parameters.
- (2) After finishing initializing all active sessions, SSetActive value is equal to SSEnable. Now the IP is ready to receive the packet of the active session. UDP data is extracted and forwarded to the user.
- (3) When user finishes all operation or needs to change some common parameters, SSEnable must be de-asserted to 0000b to terminate all sessions.
- (4) User waits until all sessions are terminated completely. SSetActive returns to 0000b.
- (5) The user sets the new common parameters. After that, user asserts some bits of SSEnable to '1' to start IP re-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 5 shows the example when enabling two sessions, i.e., session#0 and #2 in Multicast mode.

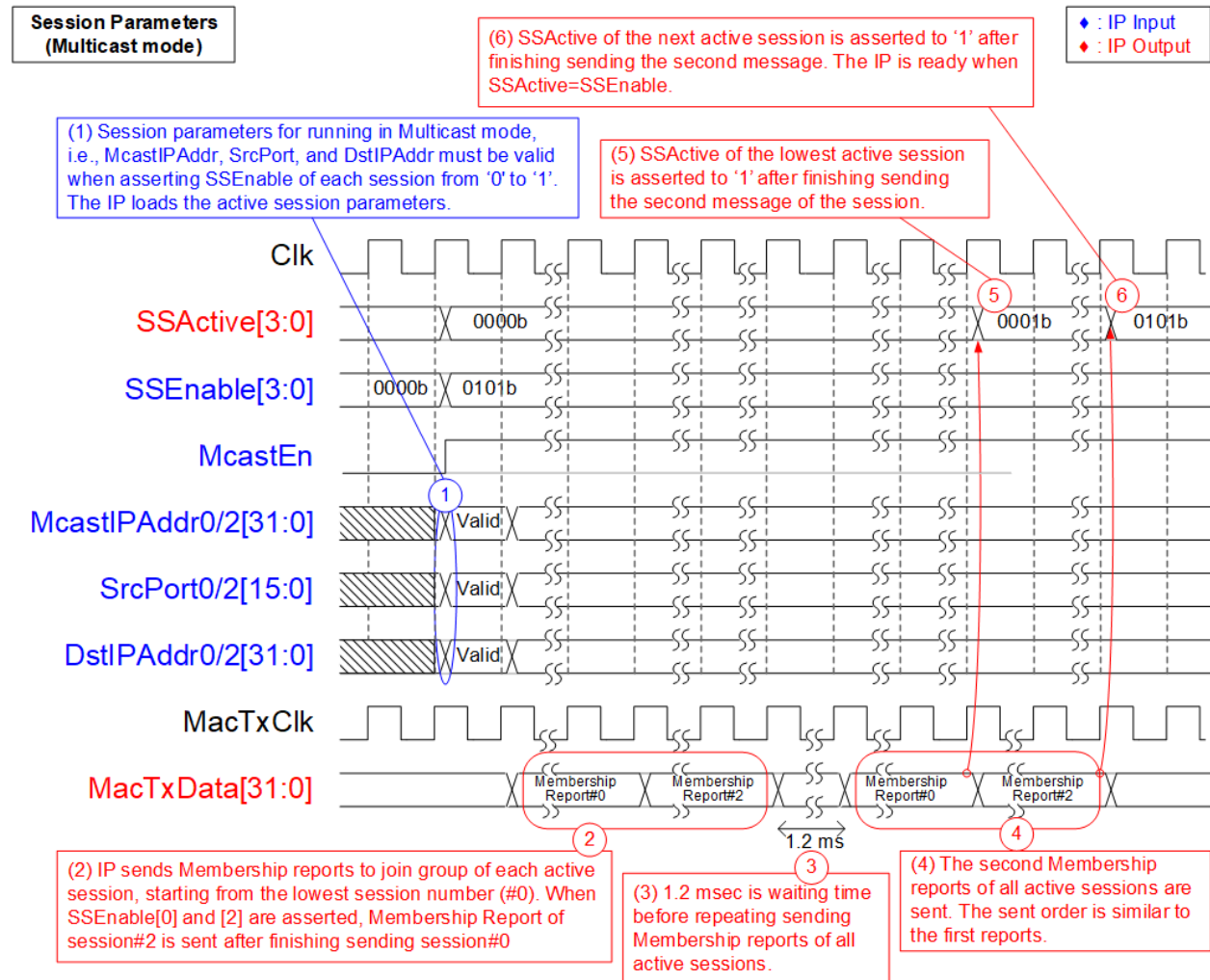


Figure 5: Multicast initialization

- (1) When some bits of SSEnable change from '0' to '1', the session parameters of the enabled session are loaded to the IP. In Multicast mode, there are three session parameters, i.e., McastIPAddr, SrcPort, and DstIPAddr that must be valid when asserting SSEnable to '1'.
- (2) IP begins the session initialization in Multicast mode by sending the first Membership report following the parameters of the active session. The IP initializes one session at a time, starting from the lowest session number. In Figure 5, SSEnable[0] and [2] are asserted. Therefore, Membership report of session#2 is sent after finishing sending session#0.
- (3) To reduce the chance of packet lost, two Membership reports are created for each session. The IP waits for 1.2 msec before sending the second reports.
- (4) IP starts sending the second Membership reports. The order for sending the second reports is similar to the sending order of the first reports, session#0 and session#2 respectively.
- (5) After finishing sending the second Membership report of the first session, SSActive of the completely initialized session (session#0) is asserted to '1'.
- (6) SSActive of the next session (session#2) is asserted to '1' after finishing sending the second Membership report. When SSActive is equal to SSEnable, the IP finishes session initialization process. The IP is ready to receive UDP packet of the active session.

Note: SSEnable can be updated only when SSActive is equal to SSEnable. Updating SSEnable during session initializing may be caused the IP maloperation.

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 6 shows the example when two sessions, i.e., session#1 and #3 are enabled in Unicast mode.

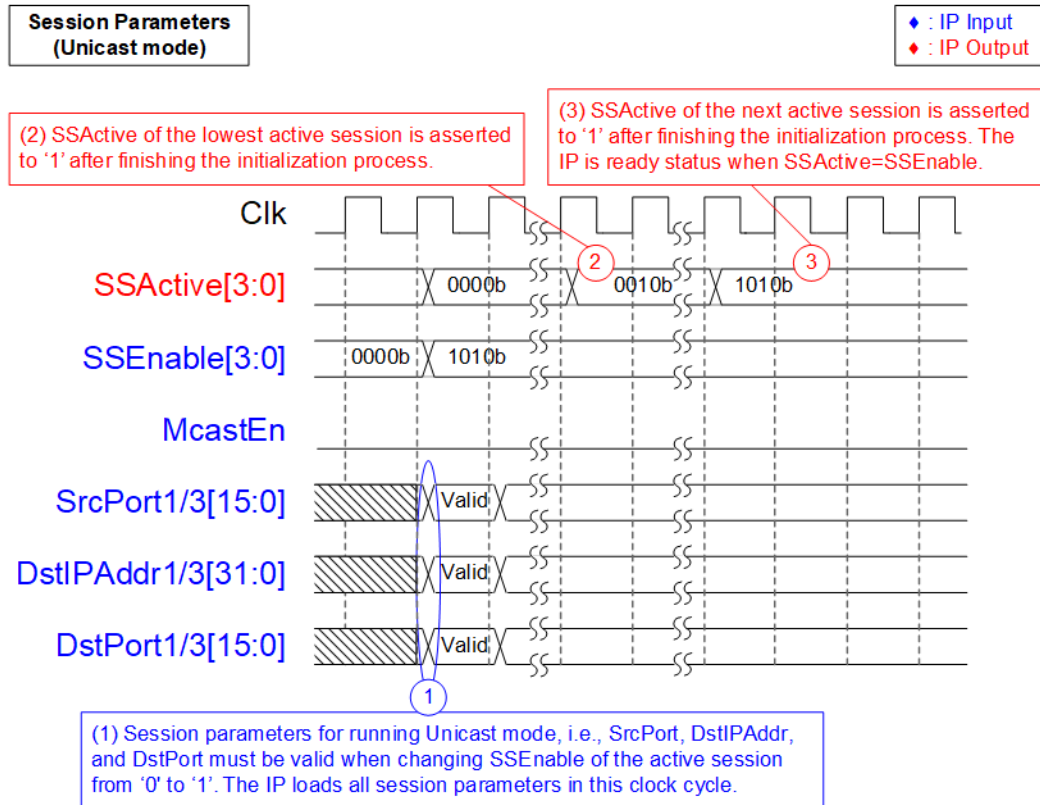


Figure 6: Unicast initialization

- (1) When some bits of SSEnable change from '0' to '1', the session parameters of the enabled session are loaded to the IP. In Unicast mode, there are three session parameters, i.e., SrcPort, DstIPAddr, and DstPort. Therefore, the session parameters must be valid when SSEnable is asserted.
- (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 Figure 6, 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 receive UDP packet of the active session.

User Data Interface

When the IP receives UDP packet from EMAC, the parameters in the header are verified. If the parameters match to the active session parameters, UDP payload data is extracted and forwarded to the user. The IP asserts error to the user if UDP checksum of a packet is not correct or EMAC asserts error status.

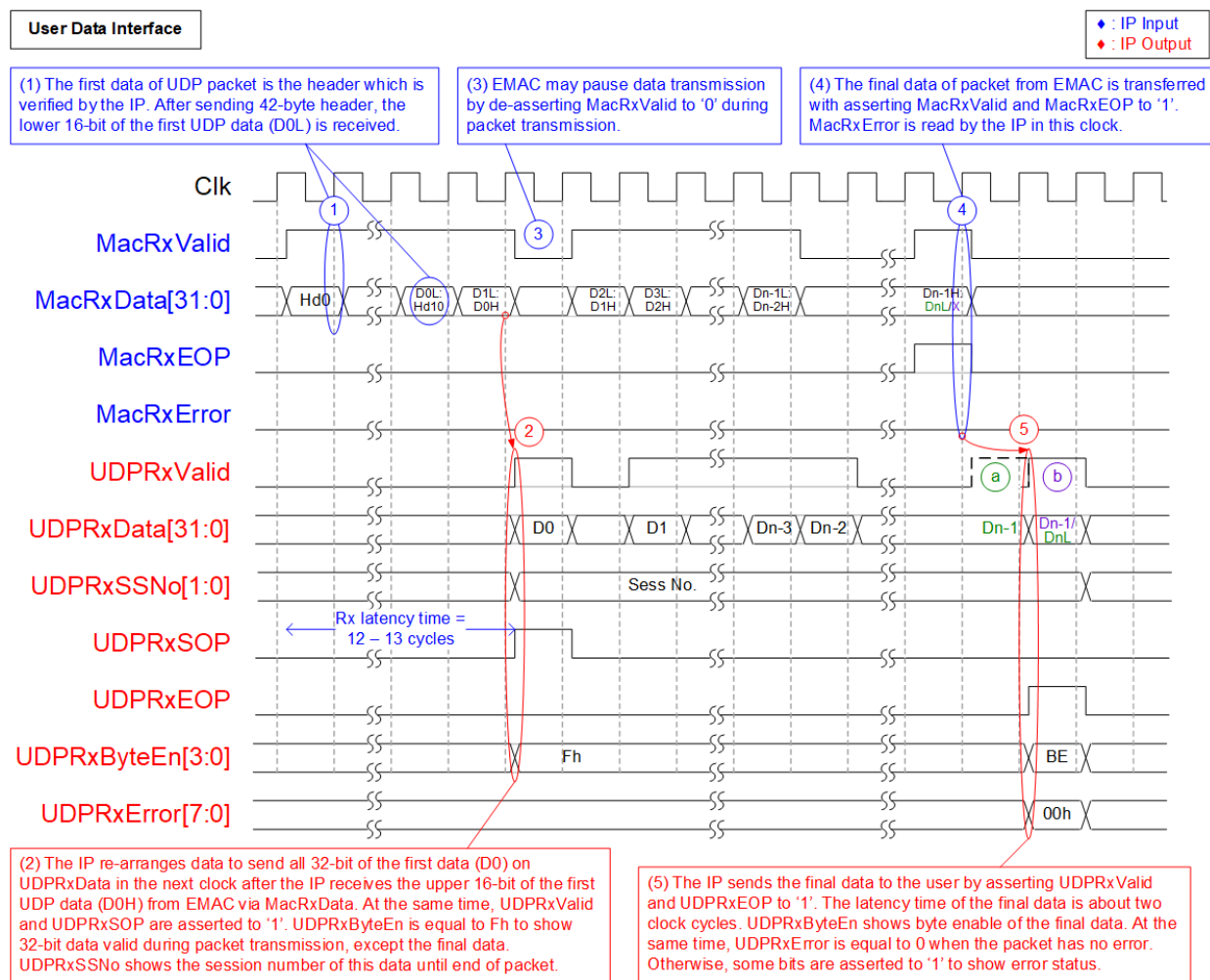


Figure 7: User data interface timing diagram

- (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 (Hd10) 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 (D0H).
- (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 final data which may be equal to 0001b (1-byte), 0011b (2-byte), 0111b (3-byte), or 1111b (4-byte). Also, UDPRxSSNo shows the session number of the received packet. The session number is latched until finishing sending all data in a packet.

Note: As shown in Figure 7, if MacRxValid is asserted continuously, the minimum latency time measured from the start of packet (Hd0 on MacRxData), output from EMAC, to the start of packet (D0 on UDPRxData), output from UDP10GRx IP, is equal to 12 clock 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' to wait for the next data from EMAC.
- (4) When EMAC sends the final 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 final data of the packet to the user by asserting UDPRxValid and UDPRxEOP to '1'. If data is valid for 1-3 bytes, 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 7, there are two timing diagrams for sending end of packet to the IP.
 - (a) When the final 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 final data is equal to 0001b or 0011b for sending remaining one-byte data or two-byte data respectively.
 - (b) When the final 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 final data is equal to 0111b or 1111b for sending three-byte data or four-byte data respectively.

Session Termination (Multicast mode)

To change the parameters of some active sessions, SSEnable of the session must be de-asserted to '0'. Figure 8 shows the example to terminate two sessions, i.e., session#0 and #2 in Multicast mode.

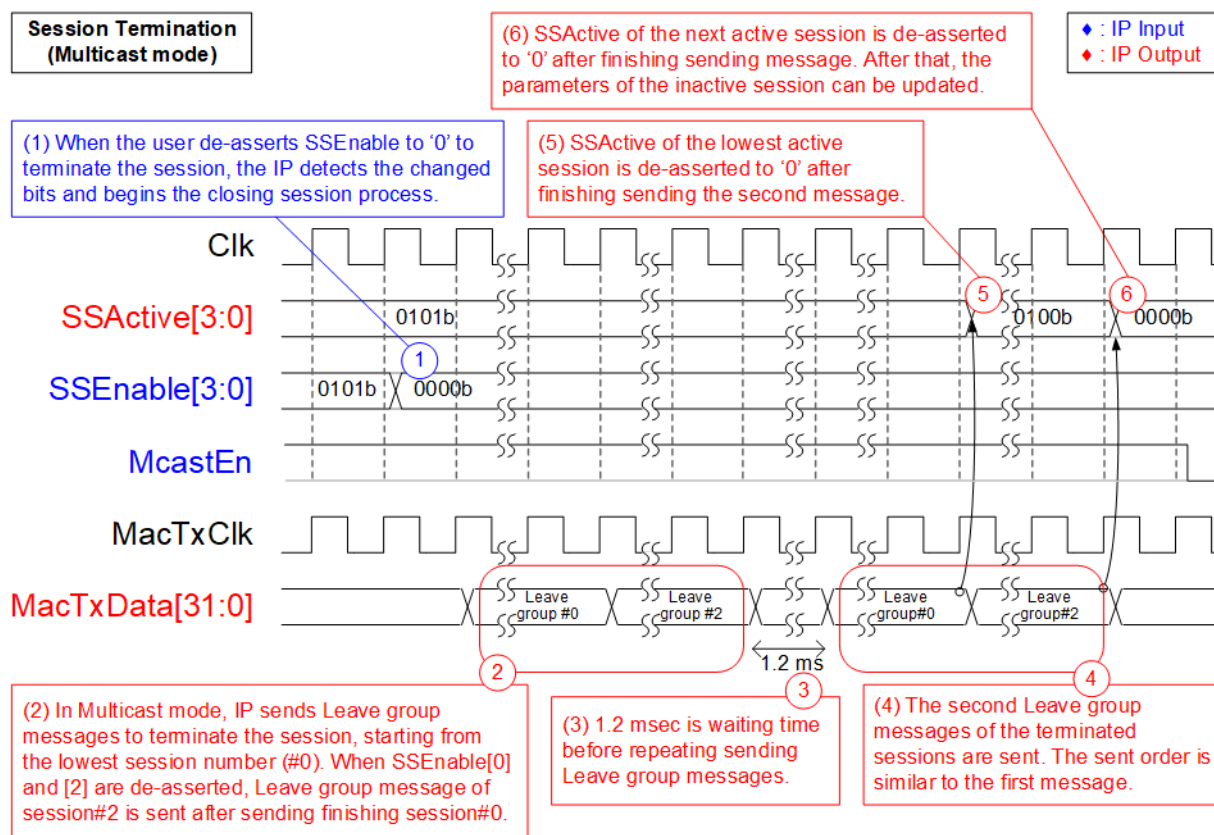


Figure 8: Multicast session termination

- (1) Some bits of SSEnable are de-asserted to '0' to terminate the session. The IP detects the changed bits and runs the process to terminate the requested session, starting from the lowest session number. In Figure 8, two sessions are requested, session#0 and session#2. Therefore, session#0 is firstly terminated.
- (2) In Multicast mode (McastEn='1'), the session is terminated by sending leave group messages. The leave group message of session#0 and session#2 are created respectively.
- (3) Similar to the session initialization, the message is created two times to reduce the chance of packet lost. Waiting time before sending the second message is equal to 1.2 msec.
- (4) The second message is created by the same order as the first message. Leave group of session#0 is sent before session#2.
- (5) After finishing sending the second leave group message of session#0, SSActive of session#0 is de-asserted to '0'.
- (6) Also, SSActive of session#2 is de-asserted to '0' after finishing sending the second leave group message of session#2. After that, the parameters of the inactive session can be updated. If SSActive is equal to 0, common parameters can be also updated.

Note: SSEnable can be updated only when SSActive is equal to SSEnable. Updating SSEnable during session initializing may be caused the IP maloperation.

Session Termination (Unicast mode)

In Unicast mode, there is no message generated to terminate the session. When the session is disabled, ARP process of the inactive session is not operated. Similar to Multicast mode, one session is terminated at a time, starting from the lowest session number, as shown in Figure 9.

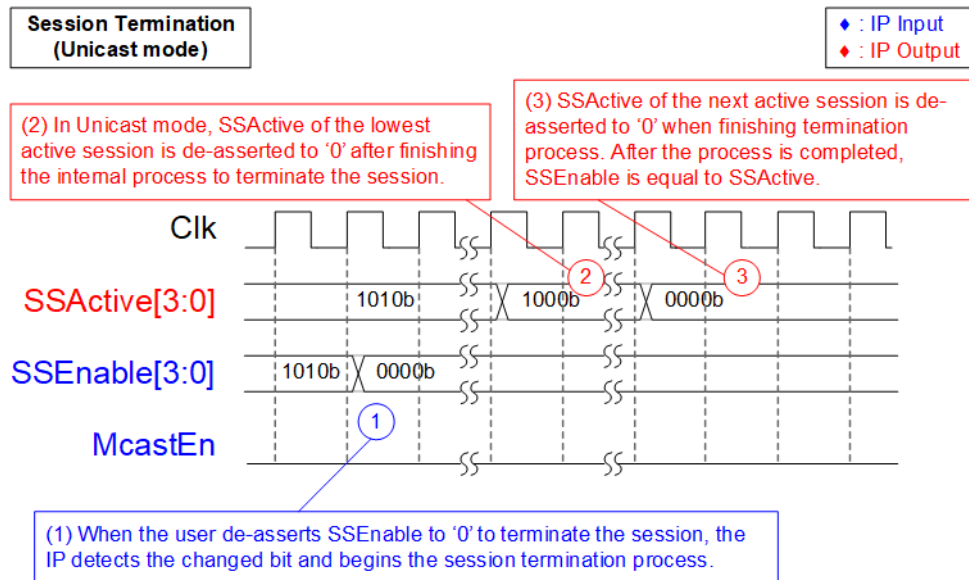


Figure 9: Unicast session termination

- (1) Some bits of SSEnable are de-asserted to '0' to terminate the session. The IP detects the changed bits and runs the termination process of the requested session, starting from the lowest session number. In Figure 9, two sessions are requested, session#1 and session#3. Therefore, session#1 is firstly terminated.
- (2) In Unicast mode (McastEn='0'), the IP runs the internal process to terminate the session. After finishing, SSetActive of the completely terminated session is de-asserted to '0'. In Figure 9, bit[1] is de-asserted and then bit[3] is de-asserted to complete the operation.
- (3) The operation is finished when SSEnable is equal to SSetActive. After that, the inactive session parameters can be updated. The IP loads the new parameters 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 ZCU102/KCU116 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 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	4-Jun-20	New release
1.1	29-Apr-21	Change clock frequency to 322.266 MHz
2.0	22-Nov-21	Support IGMPv3 and ICMP