Core10GMAC v3.0

Core10GMAC User Guide

Introduction

Core10GMAC is designed for the IEEE® 802.3-2012 specification and supports 10GBASE-R and 10-Gigabit Media-Independent Interface (XGMII). This configurable core provides the complete Media Access Control (MAC) and Physical (PHY) layer when used with a transceiver interface. The physical layer is designed to work seamlessly with the PolarFire®, PolarFire SoC, and RT PolarFire transceiver using either the Physical Medium Attachment (PMA) or 64b/66b interface modes.

This document provides information on the Core10GMAC core and the features it supports. This Core is part of the 10 Gigabit Ethernet (10GbE) subsystem, which is defined in the UG0727: PolarFire FPGA 10G Ethernet Solutions User Guide. This document provides information on how to implement 10 Gigabit Ethernet (10GbE) in PolarFire and PolarFire SoC devices. For more information, see PolarFire FPGA and PolarFire SoC FPGA Transceiver User Guide and UG0936: RT PolarFire FPGA Transceiver User Guide.

Note:
Note: The Advanced Peripheral Bus (APB) protocol standard uses the terminology “**Master**” and “**Slave**”. The equivalent Microchip terminology used in this document is “**Initiator**” and “**Target**”, respectively.

Features

Core10GMAC has the following key features:

- Configurable System Interface bus Width
- Synchronous or Asynchronous First in First Out (FIFO) Based Transmit Interface
- Synchronous or Asynchronous FIFO Based Receive Interface
- Frame Check Sequence (FCS) Insertion on Transmit, and FCS Checking on Receive
- Pad Insertion on Transmit
- Inter Frame Gap (IFG) Insertion on Transmit, while Complying with Deficit Idle Count (DIC)
- User Programmable IFG and DIC
- Configurable Preamble Size and Contents
- Pause Frame Insertion on Transmit, and Flagging on Receive

Supported Families

- PolarFire® SoC
- PolarFire
- RT PolarFire

Utilization and Performance

Resource utilization and performance data in the following tables are achieved by applying timing, synthesis, and place and route constraints. For more information, refer **Timing Constraint**, **Synthesis in Libero SoC**, **and Place-and-Route in Libero SoC.** Modify the default values of the Core10GMAC parameters such as **10G Type**, **System Data Width**, and **Core Data Width**.

The following table lists the implementation data for Core10GMAC with these configurations:**10G Type** = 10GBASE-R, **System Data Width** = 32, and **Core Data Width** = 32.

Note: 10G-BASE-R design connects to Microchip's Serializer/Deserializer (SerDes) through the Gearbox Interface.

Table 1. Core10GMAC Utilization for 10GBASE-R Design

Device Details			Resources			Utilization	Performance (MHz)	µSRAM
Family	Device	Speed Grade	4-LUT	DFF	Logic Elements	Total%		
PolarFire®	MPF300TS	-1	3844	4349	5340	1.78	I_CORE_RX_CLK – 340 I_CORE_TX_CLK – 340 I_SYS_CLK – 350 PCLK - 400	36
PolarFire SoC	MPFS250TS	-1	3844	4349	5340	2.10	I_CORE_RX_CLK – 340 I_CORE_TX_CLK – 340 I_SYS_CLK – 350 PCLK - 400	36

The following table lists the implementation data for Core10GMAC with these configurations: **10G Type** = 10GBASE-R, **System Data Width** = 64, and **Core Data Width** = 64.

Table 2. Core10GMAC Utilization for 10GBASE-R Design

Device Details			Resources			Utilization	Performance (MHz)	µSRAM
Family	Device	Speed Grade	4-LUT	DFF	Logic Elements	Total%		
PolarFire®	MPF300TS	-1	5784	6031	7624	2.55	I_CORE_RX_CLK – 340 I_CORE_TX_CLK – 340 I_SYS_CLK – 350 PCLK - 400	41
PolarFire SoC	MPFS250TS	-1	5784	6031	7624	2.99	I_CORE_RX_CLK – 340 I_CORE_TX_CLK – 340 I_SYS_CLK – 350 PCLK - 400	41

The following table lists the implementation data for Core10GMAC with these configurations: **10G Type** = XGMII,

System Data Width = 64, and **Core Data Width** = 64.

Note: XGMII design connects to Microchip's SerDes through the PCS interface using an XGMII to PCS IP (for example, CoreXAUI).

Table 3. Core10GMAC Utilization for XGMII Design

Device Details			Resources			Utilization	Performance (MHz)	µSRAM
Family	Device	Speed Grade	4-LUT	DFF	Logic Elements	Total%		
PolarFire®	MPF300TS	-1	5015	4504	6012	2.01	I_CORE_RX_CLK – 340 I_CORE_TX_CLK – 340 I_SYS_CLK – 350 PCLK - 400	41
PolarFire SoC	MPFS250TS	-1	5015	4504	6012	2.36	I_CORE_RX_CLK – 340 I_CORE_TX_CLK – 340 I_SYS_CLK – 350 PCLK - 400	41

Note: FPGA resources and performance data for the RT PolarFire family is similar to PolarFire family.

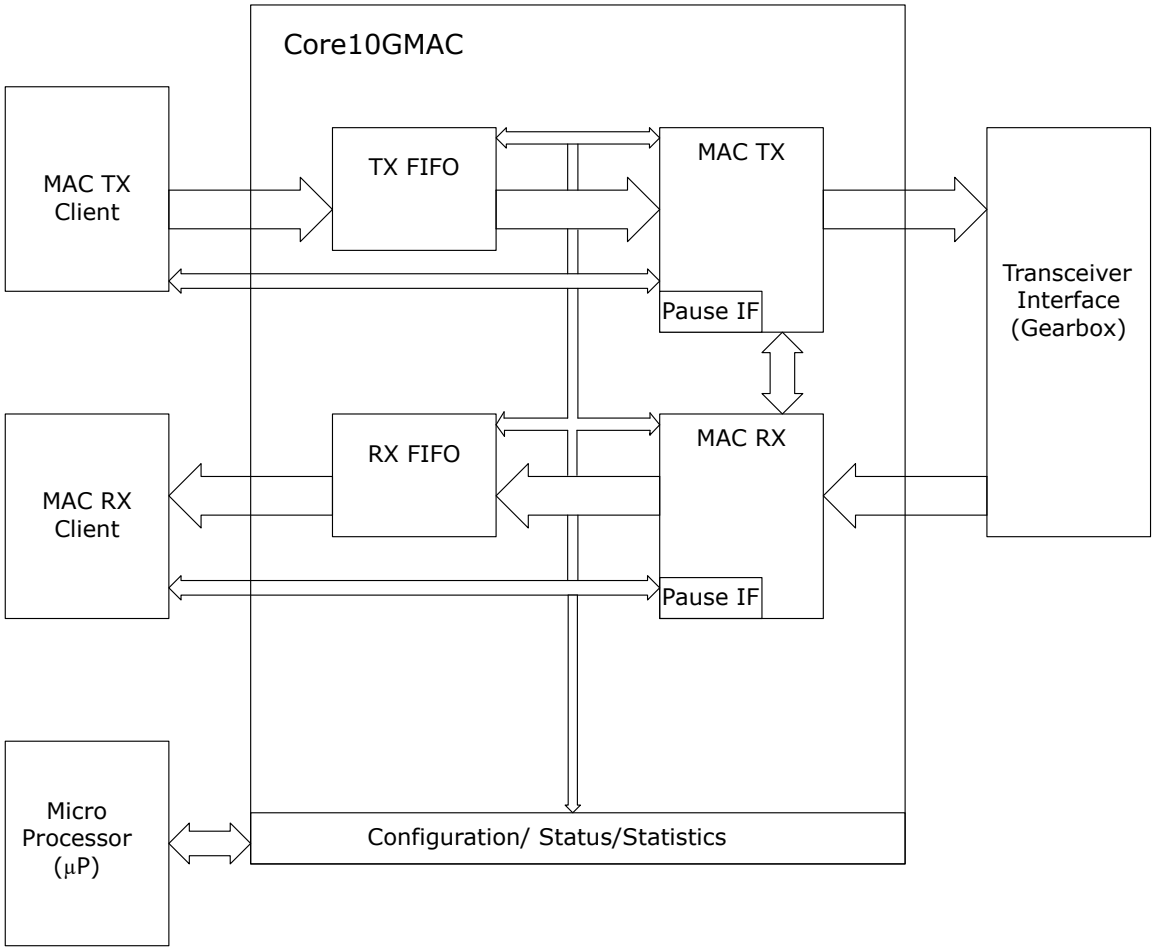
Table of Contents

1. Functional Descriptions

When you configure Core10GMAC for a 10GBASE-R design, use transceiver 64b/66b interface; and it connects directly to the MAC. 10GBASE-R supports 32-bit or 64-bit data width configuration by configuring the **System Data Width** and **Core Data Width** parameters. 10GBASE-R consists of one main block: MAC. The MAC block supports Ethernet MAC Reconciliation Sublayer (RS)/PAUSE.

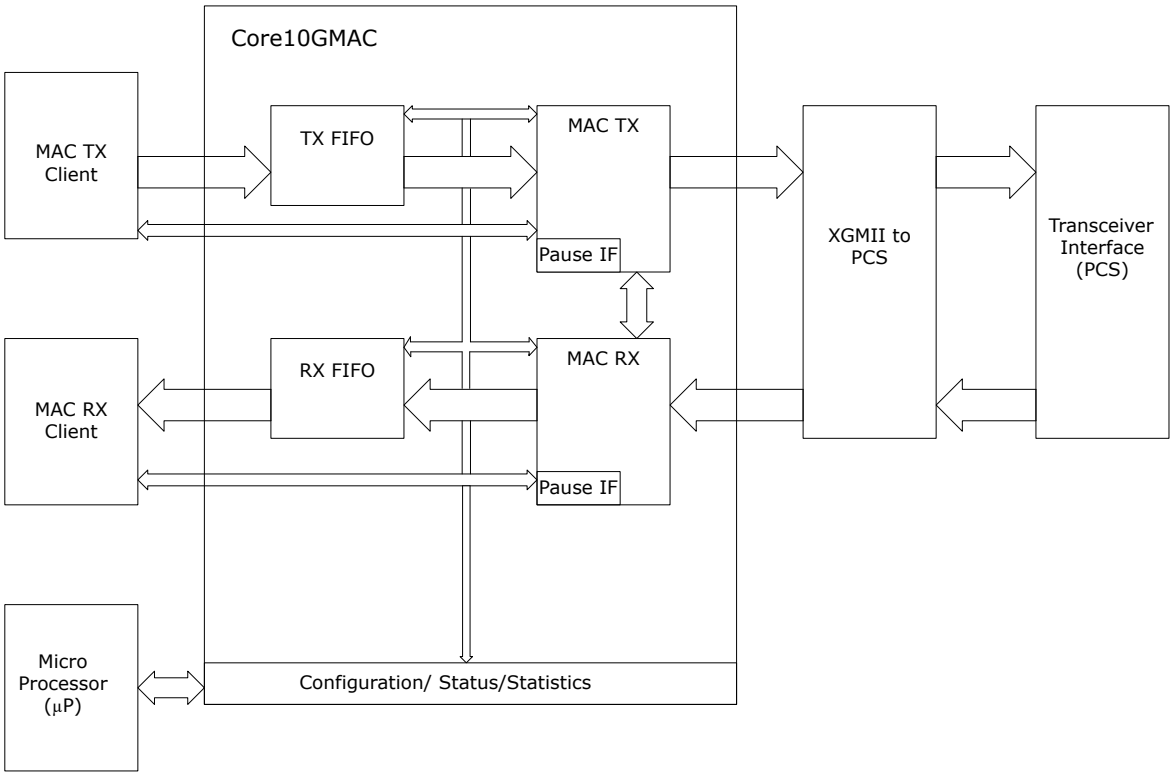
The following figure shows the Core 10GBASE-R system level block diagram.

Figure 1-1. 10GBASE-R System Level Diagram



When you configure Core10GMAC for an XGMII design, use the Physical Coding Sublayer (PCS) interface; and it connects to the MAC through an XGMII to PCS IP (for example, CoreXAUI). XGMII supports only 64-bit data width configuration by configuring the **System Data Width** and **Core Data Width** parameters. XGMII design consists of one main block: MAC. The MAC block supports Ethernet MAC RS/PAUSE.

Figure 1-2. XGMII System Level Diagram



1.1 Core10GMAC Blocks

Core10GMAC contains the following blocks.

1.1.1 Ethernet MAC/RS/PAUSE

MAC supports the following features:

- Configurable system interface bus width, which supports 32-bit or 64-bit data
- Synchronous or asynchronous FIFO based transmit interface
- Synchronous or asynchronous FIFO based receive interface
- FCS insertion on transmit, and FCS checking on receive
- Pad insertion on transmit
- IFG insertion on transmit, while complying with DIC, can be a fixed, static, or dynamic value
- User programmable IFG and DIC
- Configurable preamble size and contents, normally two words for 10GbE
- Pause frame insertion on transmit and flagging on receive
- Ethernet statistics on transmit and receive

1.1.2 Physical Coding Sub-layer

The PCS block supports the following features:

- Compliant with IEEE802.3 Clause 49, that is, PCS sublayer for 64b/66b
- 32-bit or 64-bit datapath connection to the transceiver interface

2. Operation

This section describes the operation performed by Core10GMAC.

2.1 APB Control Registers

Core10GMAC provides a 32-bit APB target interface and operates with the following register map.

2.2 Address Map

The following table lists the APB registers.

Table 2-1. Register Map

Address	Name
0x8	Transmit (Tx) Ctlr register
0x9	Receive (Rx) Ctlr registers
0xA	MAC Tx Config registers
0xB	MAC Rx Config registers
0xC	MAC Tx Static registers
0xD	MAC Rx Static registers

The following tables lists the APB registers functionality. The Address column represents PADDR[9:6] and the Offset column represents PADDR[5:2]. For example, to access Parameter Read register of sets of Rx Ctrl register, user must set PADDR[9:6] to 0x9 and PADDR[5:2] to 0x1. Action columns in the registers tables indicates access type of the registers. Access type is described as follows: RW - Read or Write RO - Read Only All the reserved bits are always read as zero and reserved bits must always be written to zero if accessed.

Note: Use the prefix 0x to represent hex number in the registers tables.

Table 2-2. Tx Ctrl Register

Address (PADDR [9:6])	Offset [PADDR [5:2]]	Register Name	Bit/s	Default	Action	Description
0x8	0x1	Tx Parameter Read	[31:4]	0	RO	**Reserved**
—	—	—	3	0	RO	**MAC Transmitter Local Loopback Enable**. Reports the value of the parameter. **MAC TX Local Loopback Enable**
—	—	—	2	0	RO	**Reserved**
—	—	—	1	0	RO	**Reserved**
—	—	—	0	0	RO	**Reserved**

Table 2-3. Rx Ctrl Registers

Address (PADDR [9:6])	Offset [PADDR [5:2]]	Register Name	Bit/s	Default	Action	Description
0x9	0x0	Rx Status Read	[31:1]	0	RO	**Reserved**
			0	0	RO	**PCS49 status** The receive status signal for 10GbE. This signal indicates that the receiver is in block lock and not in hi_ber state.
0x9	0x1	Rx Parameter Read	[31:4]	0	RO	**Reserved**
			3	0	RO	**MAC Receiver Local Loopback Enable**Reports the value of the parameter. **MAC RX Local Loopback Enable**
			2	0	RO	**Reserved**
			1	0	RO	**Reserved**
			0	0	RO	**Reserved**

Table 2-4. MAC Tx Config Registers

Address (PADDR [9:6])	Offset [PADDR [5:2]]	Register Name	Bit/s	Default	Action	Description
0xA	0x0	Tx MAC Static	[31:14]	0	RO	**Reserved**
			[13:8]	12	RW	**static_mac_tx_ifg_cnt** Configure Tx IFG Count. This signal configures the IFG value. The standard is 12, but the core supports other values. The minimum supported IFG value is 1, and the maximum is 48. System considers values less than 1 as 1, and values above 48 as 48. The signal is static and is used when the **TX IFG Count** parameter is not set to Dynamic.
			[7:5]	0	RO	**Reserved**
			4	1	RW	**static_mac_tx_ifg_dic_mode_en** Configure Tx DIC mode. This signal enables to perform IFG spacing with regards to DIC as defined by the IEEE specifications. When the signal is high, DIC is enabled, when the signal is low, DIC is disabled. The signal is static . Use it when the **TX IFG Count** parameter is set to Dynamic.

.....continued

Address (PADDR [9:6])	Offset [PADDR [5:2]]	Register Name	Bit/s	Default	Action	Description
			3	0	RW	**static_mac_tx_preamble_en** Configures the Tx core to use a customer preamble. The signal is valid when the **MAC TX Preamble** parameter is enabled.
			[2:0]	System Data Width	RW	**static_mac_tx_preamble_wcm1** Configures the size of the custom preamble, measured in words. The value is “Word Count Minus 1”, for example, 3’h0 = 1 word of preamble. The signal is valid when the **MAC TX Preamble** parameter is enabled and its initial value is set to **System Data Width** parameter.
0xA	0x1	Tx Pause MAC Address 1	[31:0]	0	RW	**pause_tx_mac_addr_lower_32bits** The lower 32 bits of the source MAC address inserted in pause frames. This signal is active when the **Tx Port/PFC** parameter is not set to Disabled.
0xA	0x2	Tx Pause MAC Address 2	[31:16]	0	RO	**Reserved**
			[15:0]	0	RW	**pause_rx_mac_addr_upper_16bits** The upper 16 bits of the source MAC address inserted in pause frames. This signal is active when the **Tx Port/PFC** parameter is not set to Disabled.
0xA	0x3	Tx Config	[31:21]	0	RO	**Reserved**
			[20:17]	0	RW	**cfg_sys_mac_tx_fifo_paf** Configure Tx FIFO programmable to almost full level. This signal configures the threshold. It is possible to change the signal dynamically.
			16	1	RW	**cfg_sys_mac_tx_en** Configure Tx enable. This configuration signal enables the MAC TX core to send frames onto the line. When this signal is asserted low, data-frames does not flow. When the signal is asserted high, data flows normally. The signal is sampled on a packet boundary, so there are no partial packets as a consequence of changing the assertion of this signal. It is possible to change the signal dynamically.

.....continued

Address (PADDR [9:6])	Offset [PADDR [5:2]]	Register Name	Bit/s	Default	Action	Description
			[15:0]	0xC000	RW	**mac_tx_max_pkt_len** Configure maximum packet length. This is the maximum configured packet length. A valid packet length is less than or equal to this signal. The signal does not influence the transmit data-path, but it is used by the transmit stats block. When [15:14] == 2'b11, the max length check is disabled. It is possible to change the signal dynamically.
0xA	0x4	Tx System 1	[31:9]	0	RO	**Reserved**
			8	0	RW	**sys_mac_tx_fcs_ins** Tx FCS insert. This signal indicates if the core must insert FCS on all the packets.
			7	0	RW	**sys_mac_tx_fcs_err** Tx FCS error. This signal indicates if the core must insert an FCS error on the packet. The FCS error is inserted by xoring the correct FCS with 0x5555_5555.
			6	0	RW	**sys_mac_tx_fcs_stomp** Tx FCS stomp. This signal indicates if the core must insert an FCS stomp. The FCS stomp is inserted by xoring the correct FCS with 0xFFFF_FFFF.
			[5:0]	0	RW	**sys_mac_tx_ifg_cnt** Tx per packet IFG. Use the signal only if **TX IFG Count** parameter is set to Dynamic.
0xA	0x5	Tx System 2	[31:0]	0	RW	**mac_tx_preamble_lower_32bits** Tx Preamble lower 32bit. Use this field only for a custom preamble when the **MAC TX Preamble** parameter is enabled and **static_mac_tx_preamble_en** bit, 3**rd** bit of Tx MAC Static Register is set to 1.
0xA	0x6	Tx System 3	[31:0]	0	RW	**mac_tx_preamble_upper_32bits** Tx preamble upper 32bits. Use this field only for a custom preamble when the **MAC TX Preamble** parameter is enabled and static_mac_tx_preamble_en bit, 3**rd** bit of Tx MAC Static Register is set to 1.

Note: Only required when parameter **System Data Width** is set to 64.

Table 2-5. MAC Rx Config Registers

Address (PADDR [9:6])	Offset [PADDR [5:2]]	Register Name	Bit/s	Default	Action	Description
0xB	0x0	Rx MAC Static	[31:4]	0	RO	**Reserved**
			3	0	RW	**mac_rx_fcs_remove** Configure Rx FCS remove. This bit configures the core to remove the FCS field. When asserted high the core strips the FCS.
			[2:0]	1	RW	**mac_rx_preamble_wcm1** Configures the size of the custom preamble, measured in words. The value is “Word Count Minus 1”, for example, 3’h0 = 1 word of preamble. The signal is active when the **MAC RX Preamble** parameter is enabled.
0xB	0x1	Rx Pause MAC Address 1	[31:0]	0	RW	**pause_rx_mac_addr_lower_32bits** The lower 32 bits of the destination MAC address helps to identify the pause uni-cast frames. This field is active when the **Rx Port/PFC** parameter is not set to Disabled.
0xB	0x2	Rx Pause MAC Address 2	[31:16]	0	RO	**Reserved**
			[15:0]	0	RW	**pause_rx_mac_addr_upper_16bits** The upper 16 bits of destination MAC address helps to identify the pause uni-cast frames. This field is active when the **Rx Port/PFC** parameter is not set to Disabled.
0xB	0x3	Rx Config	[31:18]	0	RO	**Reserved**
			17	0	RW	**sys_mac_rx_lpbk_local_en** Configure local loopback enable. The bit puts the Rx cores in Local Loopback mode. This has the effect of looping the output from the Tx MAC into the Rx MAC. The bit can be changed dynamically, but errors should be expected during the transition, especially if data is flowing when the bit is changed.
			16	1	RW	**cfg_sys_mac_rx_en** Configure Rx enable. This signal enables the reception of data from the PCS layer. When asserted data flows normally, when de-asserted the core is effectively disabled. It is possible to change the signal dynamically.

.....continued						
Address (PADDR [9:6])	**Offset** [PADDR [5:2]]	**Register Name**	**Bit/s**	**Default**	**Action**	**Description**
			[15:0]	0xC000	RW	**sys_mac_rx_max_pkt_len** Configure Rx maximum packet length. A valid packet is less than or equal to this value. If a larger packet is received it comes with O_SYS_MAC_RX_ERR and O_SYS_MAC_RX_ERR_W[2] flags. When [15:14] == 2'b11, the max length check is disabled. It is possible to change the signal dynamically.
0xB	0x4	Rx System 1	[31:1]	0	RO	**Reserved**
			0	0	RW	**sys_mac_rx_gfc** Rx global flow control. When this bit is '1', the core does not read from the receive FIFO, and when the bit is '0' the core reads normally from the receive FIFO. The response time for changes in this signal is fixed for a given core configuration but differs between configurations. The response time is between 0 and 5 clock cycles. The signal is normally tied low.

Table 2-6. MAC Tx Statistics Registers

Address (PADDR [9:6])	**Offset** [PADDR [5:2]]	**Register Name**	**Bit/s**	**Default**	**Action**	**Description**
0xC	0x0	Tx err_len_long	[31:0]	0	RO	**tx_stats_pkt_err_len_long_cnt** Number of length long errors. System flags a length long error when the packet is larger than the configured maximum, **mac_tx_max_pkt_len (bits 15 downto 0 of Tx Config Register)** This function is disabled when the **Errored at length, with long** parameter in the MAC Tx Stat Counters tab is set to disabled.
0xC	0x1	Tx err_len_check	[31:0]	0	RO	**tx_stats_pkt_err_len_check_cnt** Number of length check errors. System flags a length check error, when the LT field indicates length, and the value does not match the actual packet length. This function is disabled when the **Errored at length, with check** parameter in the MAC Tx Stat Counters tab are set to disabled.

.....continued

Address (PADDR [9:6])	Offset [PADDR [5:2]]	Register Name	Bit/s	Default	Action	Description
0xC	0x2	Tx err_len_short	[31:0]	0	RO	**tx_stats_pkt_err_len_short_cnt** Number of length short errors. System flags a length check error when the packet is smaller than the 64 bytes. This function is disabled when the **Errored at length, with short** parameter in the MAC Tx Stat Counters tab are disabled.
0xC	0x3	Tx err_fcs	[31:0]	0	RO	**tx_stats_pkt_err_fcs_cnt** Number of FCS errors. This function is disabled when the **Errored at FCS** parameter in the MAC Tx Stat Counters tab is set to disabled.
0xC	0x4	Tx err_frm	[31:0]	0	RO	**tx_stats_pkt_err_frm_cnt** Number of framing errors. System flags a framing error for the following conditions: • **sys_mac_tx_fcs_err** (7^{th} bit of Tx System 1 Register) or **sys_mac_tx_fcs_stomp** (6^{th} bit of Tx System 1 Register) is set to 1 • System interface protocol error • Transmit FIFO underruns • Transmit FIFO overflows This function is disabled when the **Errored at framing** parameter in the MAC Tx Stat Counters tab is set to disabled.
0xC	0x5	Tx Err	[31:0]	0	RO	**tx_stats_pkt_err_cnt** Number of errored packets. System disables the function when the **Errored packet** parameter in the MAC Tx Stat Counters tab is disabled.
0xC	0x6	Tx broadcast	[31:0]	0	RO	**tx_stats_pkt_broadcast_cnt** Number of broadcast packets. System disables the function when the **Broadcast packet** parameter in the MAC Tx Stat Counters tab is set to disabled.
0xC	0x7	Tx multicast	[31:0]	0	RO	**tx_stats_pkt_multicast_cnt** Number of multicast packet. System disables the function when the **Multicast packet** parameter in the MAC Tx Stat Counters tab is disabled.
0xC	0x8	Tx pause	[31:0]	0	RO	**tx_stats_pkt_pause_cnt** Number of pause packets. parameter when the **Pause packet** parameter in the MAC Tx Stat Counters tab is disabled.

.....continued						
Address (PADDR [9:6])	**Offset** [PADDR [5:2]]	**Register Name**	**Bit/s**	**Default**	**Action**	**Description**
0xC	0x9	Tx pkt_control	[31:0]	0	RO	**tx_stats_pkt_control_cnt** Number of control packets. System disables the function when the **Control packet** parameter in the MAC Tx Stat Counters tab is disabled.
0xC	0xA	Tx pkt_vlan	[31:0]	0	RO	**tx_stats_pkt_vlan_cnt** Number of VLAN packets. System disables the function when the **VLAN packet** parameter in the MAC Tx Stat Counters tab is disabled.
0xC	0xB	Tx pkt_pad	[31:0]	0	RO	**tx_stats_pkt_pad_cnt** Number of padded packets. System disables the function when the **Padded packet** parameter in the MAC Tx Stat Counters tab is disabled.
0xC	0xC	Tx pkt_ok	[31:0]	0	RO	**tx_stats_pkt_ok_cnt** Number of error free packets. System disables the function when the **Packet error free** parameter in the MAC Tx Stat Counters tab is disabled.

Table 2-7. MAC Rx Statistics Registers

Address (PADDR [9:6])	**Offset** [PADDR [5:2]]	**Register Name**	**Bit/s**	**Default**	**Action**	**Description**
0xD	0x0	Rx err_len_long	[31:0]	0	RO	**rx_stats_pkt_err_len_long_cnt** Number of length long errors. System flags a length long error when the packet is larger than the configured maximum, **mac_rx_max_pkt_len (bits 15 down to 0 of Rx Config Register)** System disables the function when the **Errored at length, with long** parameter in the MAC Rx Stat Counters tab are disabled.
0xD	0x1	Rx err_len_check	[31:0]	0	RO	**rx_stats_pkt_err_len_check_cnt** Number of length check errors. System flags a length long error when the LT field indicates length, and the value does not match the actual packet length. System disables the function when the **Errored at length, with check** parameter in the MAC Rx Stat Counter tab are disabled.
0xD	0x2	Rx err_len_short	[31:0]	0	RO	**rx_stats_pkt_err_len_short_cnt** Number of length short errors. System flags a length long error when the packet is smaller than the 64 bytes. System disables the function when the**Errored at length, with short** parameter in the MAC Rx Stat Counter tab is disabled.

.....continued

Address (PADDR [9:6])	Offset [PADDR [5:2]]	Register Name	Bit/s	Default	Action	Description
0xD	0x3	Rx err_fcs	[31:0]	0	RO	**rx_stats_pkt_err_fcs_cnt** Number of FCS errors. System disables the function when the**Errored at FCS** parameter in the MAC Rx Stat Counter tab is disabled.
0xD	0x4	Rx err_frm	[31:0]	0	RO	**rx_stats_pkt_err_frm_cnt** Number of framing errors. It indicates that the packet has flagged an error by the MAC delineation process. This happens if the PCS flags the packet in error with /E/, or if the packet is corrupt so as not to allow for proper delineation.System disables the function when the **Errored at framing** parameter in the MAC Rx Stat Counter tab is disabled.
0xD	0x5	Rx err	[31:0]	0	RO	**rx_stats_pkt_err_cnt** Number of errored packets. System disables the function when the**Errored packet** parameter in the MAC Rx Stat Counter tab is disabled.
0xD	0x6	Rx broadcast	[31:0]	0	RO	**rx_stats_pkt_broadcast_cnt** Number of broadcast packets. System disables the function when the**Broadcast packet** parameter in the MAC Rx Stat Counter tab is disabled.
0xD	0x7	Rx multicast	[31:0]	0	RO	**rx_stats_pkt_multicast_cnt** Number of multicast packet. System disables the function when the**Multicast packet**parameter in the MAC Rx Stat Counter tab is disabled.
0xD	0x8	Rx pause	[31:0]	0	RO	**rx_stats_pkt_pause_cnt** Number of pause packets. System disables the function when the **Pause packet** parameter in the MAC Rx Stat Counter tab is disabled.
0xD	0x9	Rx pkt_control	[31:0]	0	RO	**rx_stats_pkt_control_cnt** Number of control packets. System disables the function when the Graphical User Interface (GUI) **Control packet** parameter in the MAC Rx Stat Counter tab is disabled.
0xD	0xA	Rx pkt_vlan	[31:0]	0	RO	**rx_stats_pkt_vlan_cnt** Number of VLAN packets. System disables the function when the GUI **VLAN packet** parameter in the MAC Rx Stat Counter tab is disabled.

.....continued						
Address (PADDR [9:6])	**Offset** [PADDR [5:2]]	**Register Name**	**Bit/s**	**Default**	**Action**	**Description**
0xD	0xB	Rx pkt_pad	[31:0]	0	RO	**rx_stats_pkt_pad_cnt** Number of padded packets. System disables the function when the **Padded packet** parameter in the MAC Rx Stat Counter tab is disabled.
0xD	0xC	Rx pkt_ok	[31:0]	0	RO	**rx_stats_pkt_ok_cnt** Number of error free packets. System disables the function when the **Packet error free** parameter in the MAC Rx Stat Counter tab is disabled.

2.3 Nomenclature

This section provides details on a number of specific nomenclatures.

2.3.1 Static Configuration Registers

The core has a number of static configuration registers. These registers are assumed to be constant while the core is running, and the core must be reset after any of these signals change values.

All these signals are defined as static_* and are in the APB control registers. I_SYS_TX_SRESET helps to reset the Tx blocks and I_SYS_RX_SRESET helps to reset the Rx blocks.

2.3.2 Dynamic Configuration Signals

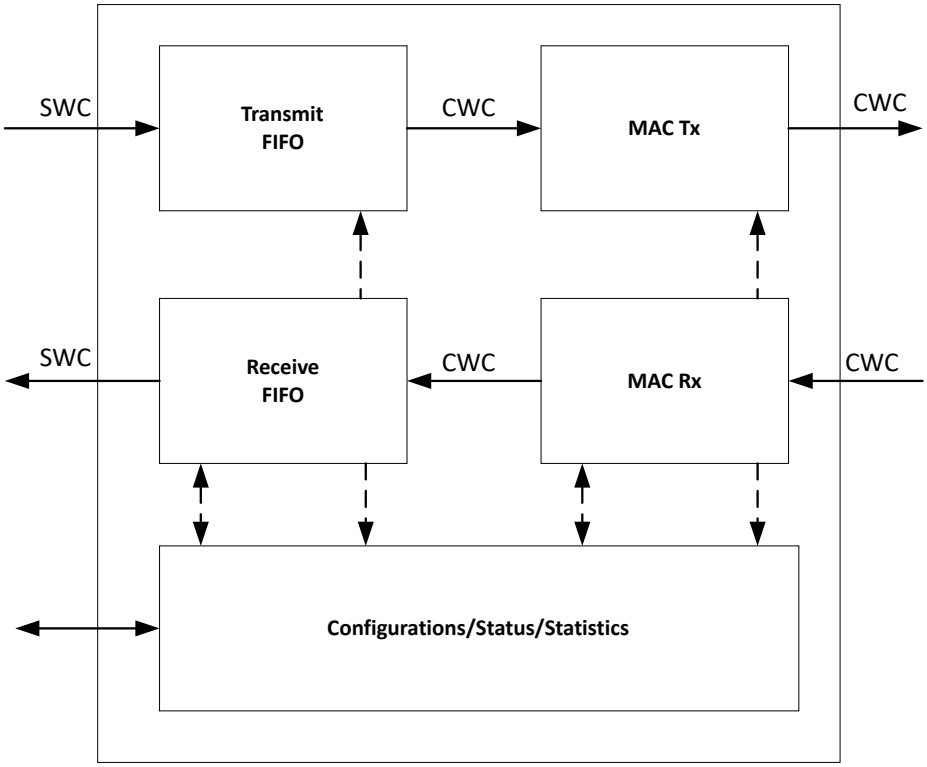
The core has a number of dynamic configuration signals. These signals can be changed at any time.

All these signals are defined as I_CFG_* and cfg_* in the APB control registers.

2.4 Ethernet MAC/RS Overview

The following figure shows a high-level architecture of the Ethernet MAC/RS.

Figure 2-1. Ethernet TX/Rx FIFO Block Diagram



2.4.1 Transmit FIFO

The Transmit FIFO decouples the user domain from the transmit clock domain. The FIFO is needed by the system to allow the MAC to adhere to the IFG insertion rules. The FIFO is implemented as an asynchronous FIFO. The user side of the FIFO performs bus protocol processing of the data delivered by the user. All badly formatted data either drops, or passes with an error. The MAC transmits the error as an FCS error.

The Transmit FIFO depth is configurable for depths of 32, 64, 128, or 256.

2.4.2 Receive FIFO

The Receive FIFO decouples the user domain from the receive clock domain. The FIFO is implemented as an asynchronous FIFO. The core controls FIFO, and the data delivery happens as a data-stream to the user side that is, the user does not have control of the FIFO flags or the read signals.

2.4.3 MAC Tx

The Transmit MAC performs the following functions:

- Reads data from the transmit FIFO
- Adds FCS
- Adds padding
- Adds preamble
- Handles FIFO underrun and overflow gracefully
- Abides by IFG
- Inserts pause frames
- Drives the statistics block
- Implements the Tx reconciliation layer

- Transmits data to PCS layer

2.4.4 MAC Rx

The Receive MAC performs the following functions:

- Receives data from the PCS layer
- Implements the Rx reconciliation layer
- Recovers the data alignment
- Calculates and checks FCS
- Extracts the preamble
- Flags bad frames
- Flags pause frames
- Drives the statistics block
- Delivers data to the receive FIFO

2.4.5 Statistics

The transmit and receive statistics are available through a statistics bus and optionally available as APB addressable counters. The user can enable which counters are to be implemented in the core by enabling respective counter fields under the MAC Tx Stat Counters and the MAC Rx Stat Counters tab in the GUI configuration.

2.5 Ethernet Interface

This section expands on the different interfaces and documents of the bus conventions.

2.5.1 Tx Dataplane

2.5.1.1 Tx Dataplane Signal Encoding

The following table lists the Tx dataplane bus protocol encoding.

Table 2-8. Tx Dataplane Bus Protocol Encoding for 64-bits (8-byte) System Data Width

Signal	Order
I_SYS_MAC_TX_DATA[63]	MSb
I_SYS_ MAC_TX_DATA [63:56]	MSB
I_SYS_ MAC_TX_DATA [7:0]	LSB
I_SYS_ MAC_TX_DATA [0]	LSb

The following table lists the associated encoding of I_SYS_MAC_TX_BC.

Table 2-9. Encoding of I_SYS_MAC_TX_BC for 64-bits (8-byte) System Data Width

I_SYS_MAC_TX_BC	Data
3’h0	I_SYS_ MAC_TX_DATA [63:56] valid
3’h1	I_SYS_ MAC_TX_DATA [63:48] valid
3’h2	I_SYS_ MAC_TX_DATA [63:40] valid
3’h3	I_SYS_ MAC_TX_DATA [63:32] valid
3’h4	I_SYS_ MAC_TX_DATA [63:24] valid
3’h5	I_SYS_ MAC_TX_DATA [63:16] valid
3’h6	I_SYS_ MAC_TX_DATA [63:8] valid
3’h7	I_SYS_ MAC_TX_DATA [63:0] valid

Table 2-10. The following table lists the Tx Dataplane Bus Protocol Encoding for 32-bits (4-byte) System Data Width

Signal	Order
I_SYS_MAC_TX_DATA [31]	MSb
I_SYS_ MAC_TX_DATA [31:24]	MSB
I_SYS_ MAC_TX_DATA [7:0]	LSB
I_SYS_MAC_TX_DATA [0]	LSb

The associated encoding of I_SYS_MAC_TX_BC is listed in the following table.

Table 2-11. The following table lists the Encoding of I_SYS_MAC_TX_BC for 32 bits (4 byte) System Data Width

I_SYS_TX_BC[1:0]	Data
2'h0	I_SYS_ MAC_TX_DATA [31:24] valid
2'h1	I_SYS_ MAC_TX_DATA [31:16] valid
2'h2	I_SYS_ MAC_TX_DATA [31:8] valid
2'h3	I_SYS_ MAC_TX_DATA [31:0] valid

2.5.1.2 Tx Dataplane Interface Errors

The Tx core performs complete error checks on the delivered data. You can introduce the following three types of errors:

- Protocol error: Errors in the transmit FIFO interface usage.
- Transmit FIFO overflow error: The transmit FIFO becomes full during the transmission of a packet. This does not happen if the user abides by the FIFO flags, but the core handles the event gracefully.
- Transmit FIFO underrun error: This event happens when the user does not deliver data fast enough.
  The Tx machine maintains protocol consistency during any of these events and flags the associated error event on the associated output signal. The errors cause the error or loss of one or more packets. The user must drive the interface so as not to introduce any of these errors.

2.5.1.3 Tx Dataplane Padding

Padding is enabled in this core; all frames are sent with a minimum frame size of 64 bytes, that is, short frames are padded. The frame can be short because of any of the following reasons:

- The packet is delivered as a short packet.
- The packet becomes short because of a FIFO over flow event. To prevent this situation, use the O_SYS_MAC_TX_FIFO_AF output signal to determine how full the transmit FIFO is.

2.5.2 Rx Dataplane

2.5.2.1 Rx Dataplane Signal Encoding

The following table lists the Rx Dataplane Bus Protocol Encoding.

Table 2-12. Rx Dataplane Bus Protocol Encoding for 64-bit (8-byte) System Data Width

Signal	Order
O_SYS_MAC_RX_DATA[63]	MSb
O_SYS_ MAC_RX_DATA [63:56]	MSB
O_SYS_ MAC_RX_DATA [7:0]	LSB
O_SYS_ MAC_RX_DATA [0]	LSb

The following table lists the associated encoding of O_SYS_MAC_RX_BC.

Table 2-13. Encoding of O_SYS_MAC_RX_BC for 64-bit (8-byte) System Data Width

O_SYS_MAC_RX_BC[2:0]	Data
3'h0	O_SYS_ MAC_RX_DATA [63:56] valid
3'h1	O_SYS_ MAC_RX_DATA [63:48] valid
3'h2	O_SYS_ MAC_RX_DATA [63:40] valid
3'h3	O_SYS_ MAC_RX_DATA [63:32] valid
3'h4	O_SYS_ MAC_RX_DATA [63:24] valid
3'h5	O_SYS_ MAC_RX_DATA [63:16] valid
3'h6	O_SYS_ MAC_RX_DATA [63:8] valid
3'h7	O_SYS_ MAC_RX_DATA [63:0] valid

Table 2-14. Rx Dataplane Bus Protocol Encoding for 32-bit (4-byte) System Data Width

Signal	Order
O_SYS_MAC_RX_DATA [31]	MSb
O_SYS_MAC_RX_DATA [31:24]	MSB
O_SYS_MAC_RX_DATA [7:0]	LSB
O_SYS_MAC_RX_DATA [0]	LSb

The following table lists the associated encoding of O_SYS_MAC_RX_BC.

Table 2-15. Encoding of O_SYS_MAC_RX_BC for 32-bit (4-byte) System Data Width

O_SYS_MAC_RX_BC [1:0]	Data
2'h0	O_SYS_ MAC_RX_DATA [31:24] valid
2'h1	O_SYS_ MAC_RX_DATA [31:16] valid
2'h2	O_SYS_ MAC_RX_DATA [31:8] valid
2'h3	O_SYS_ MAC_RX_DATA [31:0] valid

2.5.2.2 Rx Dataplane Interface Errors

The rx core performs complete error checks on the delivered data. You can introduce the following types of errors:

- Receive FIFO overflow error: The receive FIFO becomes full during the reception of a packet. This does not happen if the user abides by the FIFO flags, but the core handles the event gracefully.
- Rx packet errors: These errors assert when the error is discovered, and stay asserted until the end of packet. The source of the error can be FCS error, length error, framing error, pause frame, or an error received from the PCS layer during the packet reception.
  The Rx machine maintains protocol consistency during any of these events, and flags the associated error event on the associated output signal. The errors cause the error or loss of one or more packets. The user must drive the interface so as not to introduce any of these errors.

  The receive interface supports back pressure as a vehicle for stopping the reading of data from the receive FIFO. The receive FIFO is normally shallow, and as such this signal has limited value. If the receive FIFO overflows, the core handles the event gracefully. The core does the following on receive FIFO overflow:
- If the overflow happens in the middle of the packet, the packet is colored bad internally, the overflow flag is asserted, and the remainder of the packet is dropped, and the partial that was written to the FIFO is delivered to the user with the error indication.

- If the overflow happens at the start of the packet, the system drops the complete packet.
  The dropped data is not accounted for any of the statistics. If you want to account for all the data, then the drive FIFO in such a way so as to not loose data. You can do this by synchronising clocks and increasing the FIFO depth if required.

2.5.3 System Interface Remote Loopback

Implement a system interface remote loopback (for testing purposes) by looping the receive system bus directly to the transmit system bus.

Connect the bus signals in the following manner:

- I_SYS_MAC_TX_EN => O_SYS_MAC_RX_EN
- I_SYS_MAC_TX_SOP => O_SYS_MAC_RX_SOP
- I_SYS_MAC_TX_EOP => O_SYS_MAC_RX_EOP
- I_SYS_MAC_TX_BC => O_SYS_MAC_RX_BC
- I_SYS_MAC_TX_DATA => O_SYS_MAC_RX_DATA

Set the following register bits:

- MAC Tx Config Register bit sys_mac_tx_fcs_ins => '1'
- MAC Tx Config Register bit sys_mac_tx_fcs_err => '0'
- MAC Tx Config Register bit sys_mac_tx_fcs_stomp => '0'
- MAC Rx Config Register bit mac_rx_fcs_remove => '1'
  The connections depict the scenario where the receiver is configured to remove the Cyclic Redundancy Check (CRC). If the receiver is not configured to remove the CRC, then set the MAC Tx Config Register bit sys_mac_tx_fcs_ins to '0'.

2.5.4 Local Loopback

A local loopback (for testing purposes) can be implemented by looping the receive local loopback bus directly to the transmit local loopback bus. The MAC TX Local Loopback Enable parameter and the MAC RX Local Loopback Enable parameter must be enabled to expose the Tx\Rx local loopback interface.

Connect the bus signals in the following manner:

- O_MAC_TX_LPBK_LOCAL_CLK => I_MAC_RX_LPBK_LOCAL_CLK
- O_MAC_TX_LPBK_LOCAL_CALL => I_MAC_RX_LPBK_LOCAL_CALL
- O_MAC_TX_LPBK_LOCAL_CTRL_W => I_MAC_RX_LPBK_LOCAL_CTRL_W
- O_MAC_TX_LPBK_LOCAL_DATA_W => I_MAC_RX_LPBK_LOCAL_DATA_W

Set the following register bits:

MAC Rx Config Register bit sys_mac_rx_lpbk_local_en => '1'

Note: It is possible to change the signal dynamically, but errors must be expected during the transition, especially if data is flowing when the signal is changed.

2.5.5 Pause Interface

The core supports the transmission, reception, and procession of the pause frames. The output signals O_SYS_PAUSE_RX_PORT_XOFF or O_SYS_PAUSE_RX_PFC_XOFF_W are asserted when pause frame is received. For pause priority based flow control (pfc) , the core receives and processes the frames, but the xoff signal needs to be acted upon external to the transmit core, as the transmit MAC only has one system FIFO queue.

2.5.5.1 Pause Tx

The source MAC address for the pause frames is sourced from pause_tx_mac_addr_lower_32bits and pause_tx_mac_addr_upper_16bits registers (MAC Tx Config Registers offset 1-Tx Pause MAC Address 1 and offset 2-Tx Pause MAC Address 2 registers)

The core is enabled to send pause frames if I_CFG_PAUSE_TX_PORT_EN or I_CFG_PAUSE_TX_PFC_EN_W is asserted.

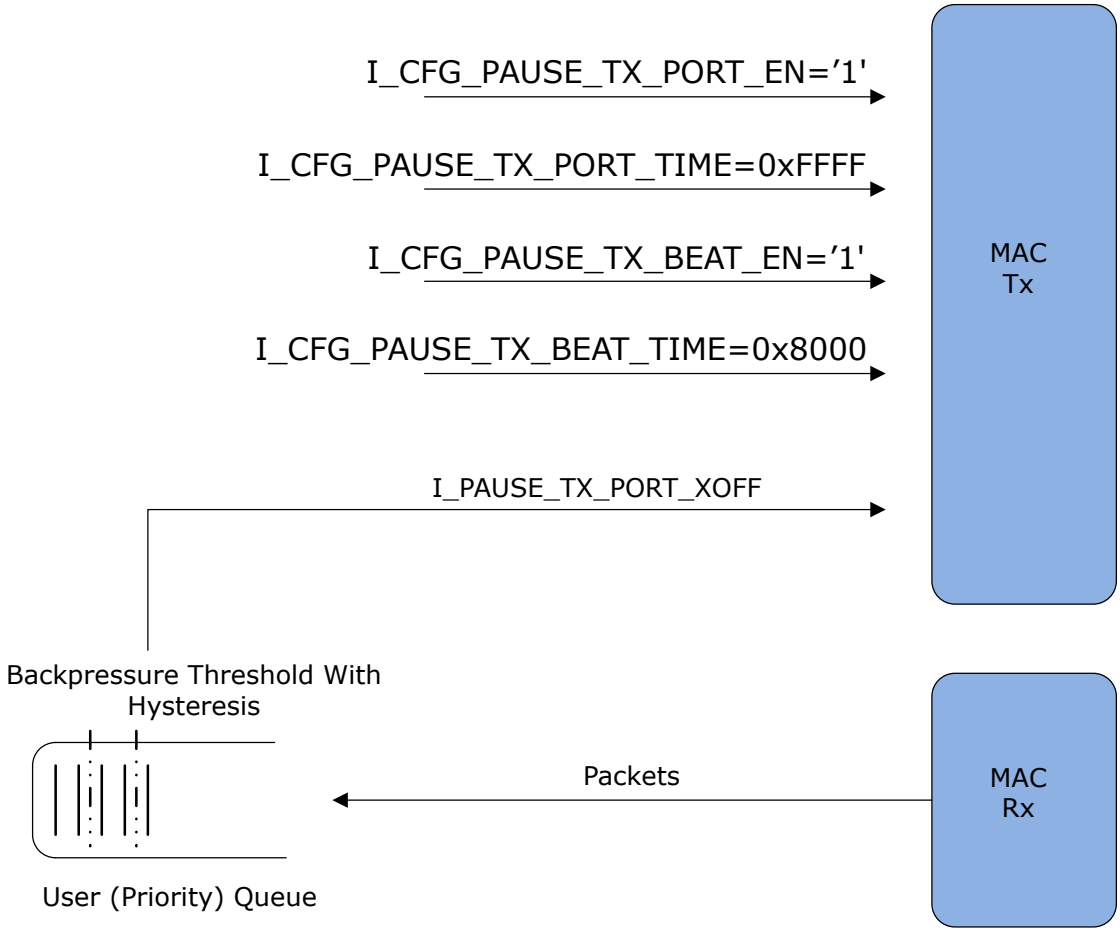
The core sends a pause frame under one of the following conditions:

- The user drives I_CFG_PAUSE_TX_SEND_STRB high
- A transition happens on I_PAUSE_TX_PORT_XOFF or I_PAUSE_TX_PFC_XOFF_W
- I_CFG_PAUSE_TX_BEAT_EN is enabled and I_CFG_PAUSE_TX_BEAT_TIME has expired
  The pause port is active when the parameter Tx Port/PFC is not set to disabled.

2.5.5.1.1 Example of Tx Configuration

The following figure shows the most common Tx configuration.

Figure 2-2. Pause Tx Example Configuration



When the user queue crosses its defined backpressure threshold, the user must assert I_PAUSE_TX_PORT_XOFF, and the MAC-Tx transmits a port pause frame with pause quanta of 16'hffff.

When the user queue goes below the backpressure threshold, I_PAUSE_TX_PORT_XOFF must de-assert, and the MAC-Tx transmits a port pause frame with pause quanta of 16'h0.

If I_PAUSE_TX_PORT_XOFF asserts continuously for more than the I_CFG_PAUSE_TX_BEAT_TIME quantas, then another pause frame is sent with pause quanta of 16'hffff.

The same usage applies to Priority based Flow Control (PFC) flows.

Normally, the configuration signals are configured as follows:

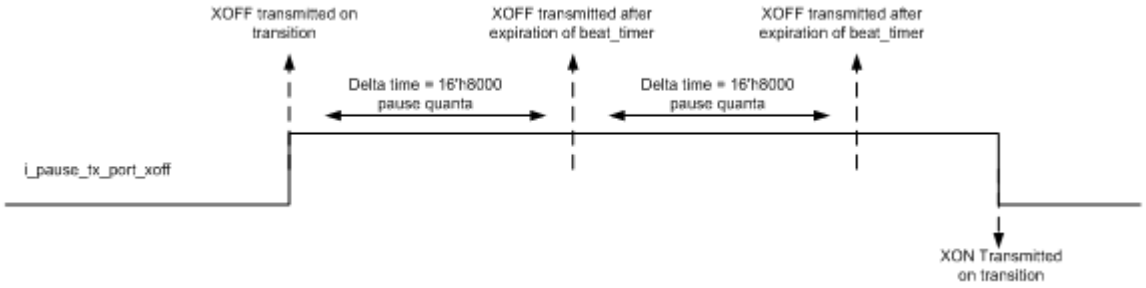
- I_CFG_PAUSE_TX_SEND_STRB = 1'b0, only used for debug
- I_CFG_PAUSE_TX_BEAT_EN = 1'b1, enable continuous sending of pause frames when in XOFF state
- I_CFG_PAUSE_TX_BEAT_TIME = 16'h8000, beat time is roughly half pause quanta time
- I_CFG_PAUSE_TX_PORT_EN = 1'b1, if port pause enabled

- I_CFG_PAUSE_TX_PORT_TIME = 16'hffff, transmitter determines when receiver turns off XOFF
- I_CFG_PAUSE_TX_PFC_EN_W = 8'hff, if PFC pause enabled
- I_CFG_PAUSE_TX_PFC_TIME_W = {8{16'hffff}}, transmitter determines when receiver turns off XOFF

2.5.5.1.2 Example of Tx Timing Diagram

The following figure shows an action diagram for the Tx timing example configuration.

Figure 2-3. Pause Tx Example Timing Diagram



A pause frame is transmitted each time there is a transition on I_PAUSE_TX_PORT_XOFF, and while I_PAUSE_TX_PORT_XOFF is asserted, a pause XOFF frame is transmitted each time I_CFG_PAUSE_TX_BEAT_TIME expires.

The beat time is internal to the core, you need to provide the expiration value by configuring I_CFG_PAUSE_TX_BEAT_TIME.

2.5.5.2 Pause Rx

The receive cores identifies pause frames by the destination address being the pause multi-cast address, or the user configured pause uni-cast address. User can configure the uni-cast addressing using the pause_rx_mac_addr_lower_32bits and pause_rx_mac_addr_upper_16bits registers (MAC Rx Config Registers offset 1 - Rx Pause MAC Address 1 and offset 2 - Rx Pause MAC Address 2) .

The core is enabled to process pause frames if I_CFG_SYS_PAUSE_RX_PORT_EN or I_CFG_SYS_PAUSE_RX_PFC_EN_W is asserted.

When a valid pause frame is received, the timer is latched and decremented, and while non-zero, the associated xoff signals (O_SYS_PAUSE_RX_PORT_XOFF or O_SYS_PAUSE_RX_PFC_XOFF_W) are asserted. O_SYS_PAUSE_RX_PORT_XOFF or O_SYS_PAUSE_RX_PFC_XOFF_W signals remain asserted until pause time out received in the pause frame or completion of pause time that is, when timer is zero. When pause frame is received, Core10GMAC does not stop the packet transmission automatically. User must stop the packet transmission by asserting I_PAUSE_TX_PORT_REQ signal when pause frame is received. If the packet transmission is in progress and I_PAUSE_TX_PORT_REQ is asserted then Core10GMAC stops packet transmission on the next packet boundary i.e Core10G MAC continues ongoing packet transmission.

2.5.6 Clocking and Resets

2.5.6.1 Clock Desciption

I_SYS_CLK: The system clock decouples the user clock domain from the core clock domains. To drive this clock either use the clock that drives I_CORE_TX_CLK or I_CORE_RX_CLK or use another completely different clock. For higher speed, this clock must be the same clock as I_CORE_TX_CLK, this provides the lowest latency.

I_CORE_TX_CLK: The Tx clock helps to transmit data. The transceivers tx clock drives this. The frequency is implementation specific. When parameter **Core Data Width** is set to 32 bit, maximum frequency of I_CORE_TX_CLK is 322.25625 MHz and when parameter **Core Data Width** is set to 64 bit, maximum frequency of I_CORE_TX_CLK is 161.1328125 MHz.

I_CORE_RX_CLK: The RX clock helps to receive data. The transceivers rx clock drives this. The frequency is implementation specific. When parameter **Core Data Width** is set to 32 bit, maximum frequency of I_CORE_RX_CLK is 322.25625 MHz and when parameter **Core Data Width** is set to 64 bit, maximum frequency of I_CORE_RX_CLK is 161.1328125 MHz.

2.5.7 Initialization

The core is initialized through its APB interface. For more details, see 2.1. APB Control Registers. The core has independent transmit and receive reset signals, but they are both referenced to the I_SYS_CLK. The core implements an internal reset staging scheme. The core indicates when it is ready on O_SYS_MAC_TX_RDY and O_SYS_MAC_RX_RDY assertions.

The Tx FIFO flag, that is, O_SYS_MAC_TX_FIFO_AF asserts 1 clock cycle after I_SYS_TX_SRESET is asserted, and stay asserted while the core is going through its reset cycle. Once the cycle is completes the flags de-asserts.

2.5.8 Statistics Vector

The MAC statistics are delivered through a statistics vector or optionally as APB registers. You can use the this vector to generate the desired Remote Network Monitoring (RMON) statistics.

The encoding of the vector for **O_SYS_MAC_TX_STATS_VECTOR** ports and The folowing table lists the **O_SYS_MAC_RX_STATS_VECTOR**. The receive and transmit vectors are symmetrical.

Table 2-16. Encoding of the Vector

Bit	Name	Description
[39:32]	mac_stats_ch	This indicates the channel for the associated stats event. This signal is all zeroes for a single instance core.
31	mac_stats_pkt_en	This signal asserts for every packet that is transmitted/received. It qualifies all the other packet statistic signals.
30	mac_stats_pkt_ok	Packet error free
29	mac_stats_pkt_pad	Padded packet
28	mac_stats_pkt_vlan	VLAN packet
27	mac_stats_pkt_control	Control packet
26	mac_stats_pkt_pause	Pause packet
25	mac_stats_pkt_multicast	Multicast packet
24	mac_stats_pkt_broadcast	Broadcast packet
23	mac_stats_pkt_err	Errored packet
[22:21]	spare	Spare
20	mac_stats_pkt_err_frm	Errored at framing
19	mac_stats_pkt_err_fcs	Errored at FCS
18	mac_stats_pkt_err_len_short	Errored at length, with short
17	mac_stats_pkt_err_len_check	Errored at length, with check
16	mac_stats_pkt_err_len_long	Errored at length, with long
[15:0]	mac_stats_pkt_length	Length of the packet

2.5.8.1 Error on Framing

An error on framing event is flagged differently between transmit and receive.

On transmit it indicates an error on the Tx FIFO or a user error insertion request, that is:

- *_fcs_err bits in the APB control registers are being asserted by the user
- System interface protocol error
- FIFO underrun

- FIFO overflow

On receive, it indicates that the packet has flagged an error by the MAC delineation process. This happens if the PCS flags the packet error with /E/, or if the packet is corrupted so as not to allow for proper delineation.

2.5.8.2 Error on Length Short

A length short error is flagged when the packet is smaller than 64 bytes.

2.5.8.3 Error on Length Check

A length check error is flagged when the LT field indicates length, and the value does not match the actual packet length.

2.5.8.4 Error on Length Long

A length long error is flagged when the packet is larger than the configured maximum, that is, mac_[tx|rx]_max_pkt_len from the APB control registers.

2.5.9 Reconciliation Sublayer

The transmit reconciliation sublayer can put the transmit core into a fault state. The following table lists the priority of the fault state generation.

Table 2-17. Priority of the Fault State Generation

Priority	Fault
0	Local
1	Remote
2	Idle

2.5.10 Latency

These latency measurements are for the core only and they do not include the latency through the SERDES.

2.5.10.1 Transmit Path Latency

The transmit path latency for XGMII type is measured from the I_SYS_MAC_TX_DATA input port of the MAC TX PACKET interface untill that data appears on the O_XGMII_TXD output port of the XGMII interface.

The transmit path latency for 10GBASE-R is measured from the I_SYS_MAC_TX_DATA input port of the MAC TX PACKET interface until that data appears on the O_PMA49_TX_GRBX_DATA output port of the 10GbE PCS interface.

Following table shows the transmit path latency for different configurations.

Table 2-18. Core10GMAC Tx Latency

10G Type	System Data Width	Core Data Width	Path		Latency (ns)	Number of Clocks	Clock Frequency (MHz)
			From	To			
10GBASE-R	64	64	I_SYS_MAC_TX_DATA	O_PMA49_TX_GRBX_DATA	304.09498	49	161.134
10GBASE-R	32	32	I_SYS_MAC_TX_DATA	O_PMA49_TX_GRBX_DATA	145.84288	47	322.266
10GBASE-R	64	32	I_SYS_MAC_TX_DATA	O_PMA49_TX_GRBX_DATA	148.94	48	322.266
XGMII	64	64	I_SYS_MAC_TX_DATA	O_XGMII_TXD	262.4	41	156.25

2.5.10.2 Receive Path Latency

The receive path latency for XGMII type is measured from the I_XGMII_RXD input port of the XGMII interface untill that data appears on the O_SYS_MAC_RX_DATA output port of the MAC RX PACKET interface.

The receive path latency for 10GBASE-R is measured from the I_PMA49_RX_GRBX_DATA input port of the 10GbE PCS interface until that data appears on the O_SYS_MAC_RX_DATA output port of the MAC RX PACKET interface.

Following table shows the transmit path latency for different configurations.

Table 2-19. Core10GMAC Rx Latency

10G Type	System Data Width	Core Data Width	Path		Latency (ns)	Number of Clocks	Clock Frequency (MHz)
			From	To			
10GBASE-R	64	64	I_PMA49_RX_GRBX_DATA	O_SYS_MAC_RX_DATA	173.76856	28	161.134
10GBASE-R	32	32	I_PMA49_RX_GRBX_DATA	O_SYS_MAC_RX_DATA	87.31012	28	322.266
10GBASE-R	64	32	I_PMA49_RX_GRBX_DATA	O_SYS_MAC_RX_DATA	93.55	30	322.266
XGMII	64	64	I_XGMII_RXD	O_SYS_MAC_RX_DATA	115.2	18	156.25

3. Interface Description

This section discusses the parameters in the Core10GMAC GUI configurator and I/O signals.

3.1 Configuration GUI Parameters

The following table lists the Core10GMAC GUI parameters for configuring the core.

Note: The Name column in the following table shows the actual parameter names used in RTL. The Description column starts with the parameter names as they appear in the Core10GMAC GUI configurator.

Table 3-1. GUI Parameters

Name	Valid Values	Default Value	Description
PERSONALITY			
TYPE_10G	10GBASE-R XGMII	10GBASE-R	**10G Type** Selects 10G Type: 0: 10GBASE-R 1: XGMII
SWC	32 64	32	**System Data Width** System word count. A word is 4-bytes, and the valid values are 32-bits (4-bytes) and 64-bits (8-bytes).
CWC	32 64	32	**Core Data Width** Core word count. Specifies the bus width of the internal core measured in words. A word is 4-bytes, and the valid values are 32-bits (4-bytes) and 64-bits (8-bytes).
MAC FEATURE			
CFG_MAC_TX_FIFO_DEPTH	32 64 128 256	32	**MAC TX FIFO Depth** The depth of the system FIFO. Normally it is set to 32 entries.
CFG_MAC_TX_PREAMBLE	0-1	0	**MAC TX Preamble** '1': Core is enabled to send custom preamble and the associated signals are active. ‘0’: Function is disabled.

.....continued

Name	Valid Values	Default Value	Description
CFG_MAC_TX_IFG_CNT	Dynamic Fixed at 1 Fixed at 2 Fixed at 3 Fixed at 4 . . . Fixed at 48	Fixed at 12	**TX IFG Count** Configure Tx IFG Count. This signal configures the IFG amount. The default value is 12, but the core also supports other values. The minimum supported IFG value is 1, and the maximum is 48. When Tx IFG Count is Dynamic, the value on sys_mac_tx_ifg_cnt (5 down to 0 bits) of the Tx system 1 register is sampled on I_SYS_MAC_TX_EOP and it is used for the associated packet.
CFG_MAC_TX_CHECK_LT	0-1	0	**MAC TX Check LT Field** '1': Core checks the LT field against the actual size of the packet. '0': Core does not check the LT field against the actual size of the packet. The checking has no consequence for the data-path, that is, it only influences the statistics reporting.
CFG_MAC_TX_LPBK_LOCAL_EN	0-1	0	**MAC TX Local Loopback Enable** '1': Core supports local loopback. '0': Core does not support local loopback.
CFG_MAC_RX_FIFO_DEPTH	32 64 128 256	32	**MAC RX FIFO Depth** The depth of the system FIFO. Normally it is set to 32 entries.
CFG_MAC_RX_PREAMBLE	0-1	0	**MAC RX Preamble** '1': Core is enabled to receive custom preamble, and the associated signals are active. '0': Function is disabled.

.....continued

Name	Valid Values	Default Value	Description
CFG_MAC_RX_CHECK_LT	0-1	0	**MAC RX Check LT Field** When '1' the core checks the LT field against the actual size of the packet. When '0' the core does not check the LT field against the actual size of the packet. The checking has consequence for the data-path and the statistics reporting.
CFG_MAC_RX_LPBK_LOCAL_EN	0-1	0	**MAC RX Local Loopback Enable** When '1' the core supports local loopback. When '0' the core does not support local loopback.
CFG_SYS_MAC_RX_GFC	0-1	0	**Rx Global Flow Control** '1': I_SYS_MAC_RX_GFC input is used. '0': The 0th bit of Rx System 1 Register-sys_mac_rx_gfc is used. When the Rx Global Flow Control is asserted high, the core does not read from the receive FIFO, and when the signal is asserted low, the core reads normally from the FIFO.
TX_MAC_STATS_PKT_ERR_LEN_LONG_CNT_EN TX_MAC_STATS_PKT_ERR_LEN_CHECK_CNT_EN TX_MAC_STATS_PKT_ERR_LEN_SHORT_CNT_EN TX_MAC_STATS_PKT_ERR_FCS_CNT_EN TX_MAC_STATS_PKT_ERR_FRM_CNT_EN TX_MAC_STATS_PKT_ERR_CNT_EN TX_MAC_STATS_PKT_BROADCAST_CNT_EN TX_MAC_STATS_PKT_MULTICASE_CNT_EN TX_MAC_STATS_PKT_PAUSE_CNT_EN TX_MAC_STATS_PKT_CONTROL_CNT_EN TX_MAC_STATS_PKT_VLAN_CNT_EN TX_MAC_STATS_PKT_PAD_CNT_EN TX_MAC_STATS_PKT_OK_CNT_EN	0-1	0	MAC TX static packet Counters Enable - Use the counters to monitor the static packet signals from the static vector output by incrementing their values on every occurrence.

.....continued			
Name	**Valid Values**	**Default Value**	**Description**
RX_MAC_STATS_PKT_ERR_LEN_LONG_CNT_EN RX_MAC_STATS_PKT_ERR_LEN_CHECK_CNT_EN RX_MAC_STATS_PKT_ERR_LEN_SHORT_CNT_EN RX_MAC_STATS_PKT_ERR_FCS_CNT_EN RX_MAC_STATS_PKT_ERR_FRM_CNT_EN RX_MAC_STATS_PKT_ERR_CNT_EN RX_MAC_STATS_PKT_BROADCAST_CNT_EN RX_MAC_STATS_PKT_MULTICASE_CNT_EN RX_MAC_STATS_PKT_PAUSE_CNT_EN RX_MAC_STATS_PKT_CONTROL_CNT_EN RX_MAC_STATS_PKT_VLAN_CNT_EN RX_MAC_STATS_PKT_PAD_CNT_EN RX_MAC_STATS_PKT_OK_CNT_EN	0-1	0	MAC RX Static Packet Counters Enable - Use the counters to monitor the static packet signals from the static vector output by incrementing their values on every occurrence.
PAUSE FEATURES			
CFG_PAUSE_TX_EN	Disabled Port Enabled PFC Enabled Port & PFC Enabled	Disabled	**Tx Port/PFC** When Disabled, the core support for pause is disabled. When Port Enabled, the core is enabled to support pause port. When PFC Enabled, the core is enabled to support pause pfc. When Port & PFC Enabled, the core is enabled to support pause port and pause pfc.
CFG_PAUSE_TX_TIMER	0-1	0	**Tx Timer Enable** ‘1’: The core is enabled to support pause timer on transmit, that is the ability to repeat send pause frames on a user defined beat. ‘0’: The function is not supported.

.....continued			
Name	**Valid Values**	**Default Value**	**Description**
CFG_PAUSE_RX_EN	Disabled Port Enabled PFC Enabled Port & PFC Enabled	Disabled	**Rx Port/PFC** When Disabled, the core support for pause is disabled. When Port Enabled, the core is enabled to support pause port. When PFC Enabled, the core is enabled to support pause pfc. When Port & PFC Enabled, the core is enabled to support pause port and pause pfc.
CFG_PAUSE_RX_CHECK_PMC	0-1	1	**Rx Check Pause Multi-Cast** ‘1’: The core is enabled to use pause multi-cast address to identify pause frames. ‘0’: The function is not supported. You can configure this parameter only if Rx Port/PFC parameter are not disabled.
CFG_PAUSE_RX_CHECK_PUC	0-1	0	**Rx Check Pause Unicast-Cast** ‘1’: The core is enabled to use the pause unicast-cast address to identify pause frames. ‘0’: The function is not supported. You can configure this parameter only if Rx Port/PFC parameter is not disabled.
APB Timeout			
CFG_CPU_ASYNC_TO_EN	0-1	0	APB Timeout Enable. ‘1’: The core is enabled to use timeout value configured for the APB Timeout Count parameter to release PREADY of the APB interface. ‘0’: The core ignores the APB Timeout Count parameter value.

.....continued			
Name	**Valid Values**	**Default Value**	**Description**
CFG_CPU_ASYNC_TO_CNT	0-255	80	APB Timeout Count. This parameter configures timeout value When APB Timeout Enable is '1'. If PREADY of the APB interface remains low for the timeout value, then it is released. To count the timeout value, PCLK of APB interface is used.

3.2 I/O Signals

The following table lists the clock ports for Core10GMAC.

Table 3-2. Core10GMAC Clock Ports

Name	Width	Direction	Description
I_SYS_CLK	1	Input	Clock for the system side clock domain
I_CORE_TX_CLK	1	Input	Transmit clock for the line side clock domain
I_CORE_RX_CLK	1	Input	Receive clock for the line side clock domain
O_MAC_TX_LPBK_LOCAL_CLK	1	Output	MAC Tx local loopback clock port. This port must be connected to the I_MAC_RX_LPBK_LOCAL_CLK equivalent input clock port of the MAC Rx. This port is visible to you when the MAC Tx Local Loopback Enable parameter is enabled
I_MAC_RX_LPBK_LOCAL_CLK	1	Input	MAC Rx local loopback clock port. This port must be connected to an O_MAC_TX_LPBK_LOCAL_CLK equivalent input clock port of MAC Tx. This port is visible to the user when the MAC Rx Local Loopback Enable parameter is enabled
PCLK	1	Input	APB clock

The following table lists the input/output ports of Core10GMAC. The Clock Domain column in the table indicates the input ports and the output ports expected to be synchronized in the respective clock domain.

Table 3-3. Core10GMAC Input/Output Ports

Name	Clock Domain	Width	Direction	Description
SYSTEM Resets				
I_SYS_TX_SRESET	I_SYS_CLK	1	Input	Active high synchronous reset of the transmit core with reference to the I_SYS_CLK domain.
I_SYS_RX_SRESET	I_SYS_CLK	1	Input	Active high synchronous reset of the receive core with reference to the I_SYS_CLK domain.
CORE Resets				

.....continued

Name	Clock Domain	Width	Direction	Description
O_CORE_TX_SRESET	I_CORE_TX_CLK	1	Output	The internally generated active high reset signal. You can use this signal to reset other modules.
O_CORE_RX_SRESET	I_CORE_RX_CLK	1	Output	The internally generated active high reset signal. You can use this signal to reset other modules.
MAC TX PACKET INTERFACE				
O_SYS_MAC_TX_RDY	I_SYS_CLK	1	Output	Tx ready. This active high signal is de-asserted when the transmit core is going through reset and initialization. When asserted, the core is ready to transmit data. Drops writes to the transmit FIFO when the signal de-asserts. **Note**: When this signal is de-asserts, the transmit FIFO flags assert.
O_SYS_MAC_TX_FAULT	I_SYS_CLK	1	Output	Tx fault. This signal indicates that the transmit MAC is in an RS Fault state, and the system drops the packets written to the transmit FIFO. This signal does not have any interaction with the FIFO flags.
O_SYS_MAC_TX_FIFO_AF	I_SYS_CLK	1	Output	Tx FIFO almost full flag. This signal indicates that the transmit FIFO is almost full. When the signal transitions from de-assertion to assertion the FIFO has space for four more entries.
I_SYS_MAC_TX_EN	I_SYS_CLK	1	Input	Tx enable. This signal helps to qualify the writes to the Tx FIFO. The value of the other Tx data path input signals is immaterial when this signal does not assert.
I_SYS_MAC_TX_SOP	I_SYS_CLK	1	Input	Tx start of packet. This signal indicates the start of packet event.
I_SYS_MAC_TX_EOP	I_SYS_CLK	1	Input	Tx end of packet. This signal indicates the end of packet event.
I_SYS_MAC_TX_BC	I_SYS_CLK	8	Input	Tx end of packet byte count. This signal indicates the number of valid byte entries in the transfer. For more information about formatting, see **Tx Dataplane Signal Encoding**. The signal is sampled when I_SYS_MAC_TX_EN and I_SYS_MAC_TX_EOP asserts. **Note:** The core uses only the [log2(SWC*4)-1:0] bits and ignores the remaining signals.
I_SYS_MAC_TX_DATA	I_SYS_CLK	32 or 64	Input	Tx data. This signal contains the data for transfer. This signal is sampled when I_SYS_MAC_TX_EN asserts. For more information about formatting, see **Tx Dataplane Signal Encoding**.

.....continued

Name	Clock Domain	Width	Direction	Description
O_SYS_MAC_TX_ERR_BUS_PROTOCOL	I_SYS_CLK	1	Output	Transmit bus protocol error. This signal asserts for one cycle when there is an error with the user protocol on the transmit data plane interface, for example, if two SOP are received without an End Of Packet (EOP) in between.
O_SYS_MAC_TX_ERR_FIFO_OVERFLOW	I_SYS_CLK	1	Output	Transmit FIFO overflow error. This signal asserts for one cycle on a Tx FIFO overflow event. This does not happen if the user abides by the FIFO flags, but the core handles the event gracefully.
O_SYS_MAC_TX_ERR_FIFO_UNDERRUN	I_SYS_CLK	1	Output	Transmit FIFO underrun error. This signal asserts for one cycle on a Tx FIFO underrun event. The core plays out /E/ events while the FIFO is in underrun. The event happens when the user does not deliver data fast enough.
O_SYS_MAC_TX_STATS_VECTOR	I_SYS_CLK	40	Output	The transmit statistics bus. For more information about encoding, see **Statistics Vector**.
MAC RX PACKET INTERFACE				
I_SYS_MAC_RX_GFC	I_SYS_CLK	1	Input	Rx global flow control. When this signal is transitions to high, the core does not read from the receive FIFO, and when the signal asserts low, the core reads normally from the FIFO. The response time to changes in this signal is fixed for a given core configuration but differs between configurations. The response time is between 0 and 5 clock cycles. Normally, the signal is normally tied low. This signal is only available when the **Rx Global Flow Control** parameter is enabled, otherwise the sys_mac_rx_gfc (0th bit of Rx System 1 Register,) is used.
O_SYS_MAC_RX_RDY	I_SYS_CLK	1	Output	Rx core ready. This signal transitions to low while the receive core is going through reset and initialization. When asserts high, the core is ready to function.
O_SYS_MAC_RX_EN	I_SYS_CLK	1	Output	Rx enable. This signal helps to qualify the received data. The values of the other Rx data path output signals are undefined when this signal asserts.
O_SYS_MAC_RX_SOP	I_SYS_CLK	1	Output	Rx start of packet. This signal indicates the start of packet event. **Note**: This signal always drivens low when O_SYS_MAC_RX_EN de-asserts.
O_SYS_MAC_RX_EOP	I_SYS_CLK	1	Output	Rx end of packet. This signal indicates the end of packet event. **Note:** This signal always drivens low when O_SYS_MAC_RX_EN de-asserts.

.....continued

Name	Clock Domain	Width	Direction	Description
O_SYS_MAC_RX_ERR	I_SYS_CLK	1	Output	Rx error. This signal indicates an error with the associated packet. The signal asserts when the error is discovered and stay asserted until O_SYS_MAC_RX_EOP asserts. The source of the error can be FCS error, length error, framing error, pause frame, or an error received from the PCS layer during the packet reception. **Note:** Always this signal drivens low when O_SYS_MAC_RX_EN de-asserts.
O_SYS_MAC_RX_ERR_W[7]	I_SYS_CLK	1	Output	Not used.
O_SYS_MAC_RX_ERR_W[6]	I_SYS_CLK	1	Output	Rx frame error. This signal indicates a PCS error or packet framing error with the packet. The signal asserts when the error is discovered and stay asserted until O_SYS_MAC_RX_EOP asserts. When this signal asserts, O_SYS_MAC_RX_ERR also asserts.
O_SYS_MAC_RX_ERR_W[5]	I_SYS_CLK	1	Output	Rx CRC error. This signal indicates an FCS error with the associated packet. The signal asserts only when O_SYS_MAC_RX_EOP asserts. When this signal asserts, O_SYS_MAC_RX_ERR also asserts.
O_SYS_MAC_RX_ERR_W[4]	I_SYS_CLK	1	Output	Rx FCS stomp error. This signal indicates an FCS stomp with the associated packet. The signal assert only when O_SYS_MAC_RX_EOP transitions to assert. When this signal asserts, O_SYS_MAC_RX_ERR also asserts. **Note:** Stomped packet is also flagged as a CRC errored packet. Hence, you can ignore the signal.
O_SYS_MAC_RX_ERR_W[3]	I_SYS_CLK	1	Output	Rx short frame error. This signal indicates that the packet is less than 64 bytes long. The signal assert only when O_SYS_MAC_RX_EOP asserts. When this signal asserts, O_SYS_MAC_RX_ERR also asserts.
O_SYS_MAC_RX_ERR_W[2]	I_SYS_CLK	1	Output	Rx long frame error. This signal indicates that the packet is larger than sys_mac_rx_max_pkt_len (lower 16 bits of Rx Config Register). The signal asserts when the length exceeds the maximum configured and stays asserted until O_SYS_MAC_RX_EOP asserts. When this signal asserts, O_SYS_MAC_RX_ERR also asserts.

.....continued				
Name	**Clock Domain**	**Width**	**Direction**	**Description**
O_SYS_MAC_RX_ERR_W[1]	I_SYS_CLK	1	Output	Rx length/type error. This signal indicates a problem with the LT field, that is the field indicates packet length, but does not match the actual packet length. When this signal asserts, O_SYS_MAC_RX_ERR also asserts. This signal is active when the **MAC RX Check LT Field** parameter is enabled.
O_SYS_MAC_RX_ERR_W[0]	I_SYS_CLK	1	Output	Rx pause frame error. This signal indicates that the packet has been flagged by the core as a pause frame. The signal asserts only when O_SYS_MAC_RX_EOP asserts. When this signal asserts, O_SYS_MAC_RX_ERR also asserts. This signal is active when the **Rx Port/PFC** parameter is not set to Disabled.
O_SYS_MAC_RX_BC	I_SYS_CLK	8	Output	Rx byte count. This signal indicates the number of valid byte entries in the transfer. For more information about formatting, see **Rx Dataplane Signal Encoding**. **Note:** The core uses only the [log2(SWC*4)-1:0] bits and ignores the remaining signals.
O_SYS_MAC_RX_PREAMBLE	I_SYS_CLK	64	Output	Rx preamble. This signal contains the receive preamble. The signal is valid when O_SYS_MAC_RX_EN and O_SYS_MAC_RX_SOP assert.
O_SYS_MAC_RX_DATA	I_SYS_CLK	SWC*32	Output	Rx data. This signal contains the received data. The signal is only valid when O_SYS_MAC_RX_EN asserts. For more information about formatting, see **Rx Dataplane Signal Encoding**.
O_SYS_MAC_RX_ERR_FIFO_OVERFLOW	I_SYS_CLK	1	Output	Receive FIFO overflow error. This signal asserts for one cycle on an Rx FIFO overflow event. This happens only if I_SYS_CLK is not running fast enough. The core corrupts a packet, which is presently being written to the FIFO, and it completely drops a new packet while the FIFO is full. As a result, you continue to see consistent packet boundaries on the system side.
O_SYS_MAC_RX_STATS_VECTOR	I_SYS_CLK	40	Output	The receive statistics bus. For more information about encoding, see **Statistics Vector**.
MAC LOOPBACK INTERFACE				
O_MAC_TX_LPBK_LOCAL_CALL	O_MAC_TX_LPBK_LOCAL_CLK	32	Output	Local loopback ports. Connect to equivalent input signals on MAC Rx. This signal is active when the **MAC TX Local Loopback Enable** parameter is enabled.

.....continued				
Name	**Clock Domain**	**Width**	**Direction**	**Description**
O_MAC_TX_LPBK_LOCAL_CTRL_W	O_MAC_TX_LPBK_LOCAL_CLK	32 or 64	Output	Local loopback ports. Connect to equivalent input signals on MAC Rx. This signal is active when the **MAC TX Local Loopback Enable** parameter is enabled.
O_MAC_TX_LPBK_LOCAL_DATA_W	O_MAC_TX_LPBK_LOCAL_CLK	32 or 64	Output	Local loopback ports. Connect to equivalent input signals on MAC Rx. This signal is active when the **MAC TX Local Loopback Enable** parameter is enabled.
I_MAC_RX_LPBK_LOCAL_CALL	I_MAC_RX_LPBK_LOCAL_CLK	32	Input	Local loopback ports. Connect to equivalent input signals on MAC Tx. This signal is active when the **MAC RX Local Loopback Enable** parameter is enabled.
I_MAC_RX_LPBK_LOCAL_CTRL_W	I_MAC_RX_LPBK_LOCAL_CLK	32 or 64	Input	Local loopback ports. Connect to equivalent input signals on MAC Tx. This signal is active when the **MAC RX Local Loopback Enable** parameter is enabled.
I_MAC_RX_LPBK_LOCAL_DATA_W	I_MAC_RX_LPBK_LOCAL_CLK	32 or 64	Input	Local loopback ports. Connect to equivalent input signals on MAC Tx. This signal is active when the **MAC RX Local Loopback Enable** parameter is enabled.
TX PAUSE SIGNALS				
I_CFG_PAUSE_TX_SEND_STRB	I_CORE_TX_CLK	1	Input	A pulse on this signal initiates the transmission of a pause frame. This is a backdoor signal to allow software to force the transmission of a pause frame. The signal is active when the **Tx Port/PFC** parameter is not set to Disabled.
I_CFG_PAUSE_TX_BEAT_EN	I_CORE_TX_CLK	1	Input	When asserted, the core continuously generates and sends pause frames. The signal is enabled when the **Tx Timer Enable** parameter is enabled and the **Tx Port/PFC** parameter is not set to disabled.
I_CFG_PAUSE_TX_BEAT_TIME	I_CORE_TX_CLK	16	Input	The time interval that is used for the transmission beat. The timer is measured in 512 UI increments. The signal is enabled when the **Tx Timer Enable** parameter is enabled and the **Tx Port/PFC** parameter is not set to Disabled.
I_CFG_PAUSE_TX_PORT_EN	I_CORE_TX_CLK	1	Input	When asserted, the core sends pause-port frames. The signal is enabled when the **Tx Port/PFC** parameter is not set to Disbaled.
I_CFG_PAUSE_TX_PORT_TIME	I_CORE_TX_CLK	16	Input	The timer value inserted in the pause-port frames. The signal is enabled when the **Tx Port/PFC** parameter is not set to Disabled.

.....continued

Name	Clock Domain	Width	Direction	Description
I_CFG_PAUSE_TX_PFC_EN_W	I_CORE_TX_CLK	8	Input	When asserted, the core sends pause-pfc frames. I_CFG_PAUSE_TX_PFC_EN_W[0] is associated with priority 0, I_CFG_PAUSE_TX_PFC_EN_W[1] is associated with priority 1, and so on. The signal is enabled when the **Tx Port/PFC** parameter is not set to Disabled.
I_CFG_PAUSE_TX_PFC_TIME_W	I_CORE_TX_CLK	128	Input	The timer value inserted in the pause-port frames. Sixteen bits are associated with each priority. I_CFG_PAUSE_TX_PFC_TIME_W [15:0] is associated with priority 0, I_CFG_PAUSE_TX_PFC_TIME_W [31:16] is associated with priority 1, and so on. The signal is enabled when the **Tx Port/PFC** parameter is not set to disabled.
I_PAUSE_TX_PORT_XOFF	I_CORE_TX_CLK	1	Input	Data-plane signal requesting pause-port. A transition on this signal causes the transmission of a pause-port frame. When this signal is low, the timer is sent as 0, when high, the timer is sent as I_CFG_PAUSE_TX_PORT_TIME. The signal is enabled when the **Tx Port/PFC** parameter is not set to disabled and when the I_CFG_PAUSE_TX_PORT_EN parameter asserts.
I_PAUSE_TX_PFC_XOFF_W	I_CORE_TX_CLK	8	Input	Data-plane signal requesting pause-pfc. A transition on this signal causes the transmission of a pause-port frame. I_PAUSE_TX_PFC_XOFF_W[0] is associated with priority 0, I_PAUSE_TX_PFC_XOFF_W[1] is associated with priority 1, and so on. When I_PAUSE_TX_PFC_XOFF_W[x] (x ranges from 0 to 7) is low, the associated priority timer is sent as 0, when high, the associated timer is sent as I_CFG_PAUSE_TX_PFC_TIME_W[x+:16] (x ranges from 0 to 7).The signals is enabled when the **Tx Port/PFC** parameter is not set to disabled and when the I_CFG_PAUSE_TX_PFC_EN_W[x] (x ranges from 0 to 7) parameter asserts.
I_PAUSE_TX_PORT_REQ	I_CORE_TX_CLK	1	Input	When asserted, the transmit MAC is requested to stop packet transmission. This signal is sourced by the pause-port receive process.
O_PAUSE_TX_PORT_ON	I_CORE_TX_CLK	1	Output	Indicates that the transmit MAC has halted packet transmission due to request on I_PAUSE_TX_PORT_REQ.
RX PAUSE SIGNALS				

.....continued				
Name	**Clock Domain**	**Width**	**Direction**	**Description**
I_SYS_PAUSE_RX_SRESET	I_SYS_CLK	1	Input	Reset of the pause Rx block. This signal is active when the **Rx Port/PFC** parameter is not set to disabled.
I_CFG_SYS_PAUSE_RX_PORT_EN	I_SYS_CLK	1	Input	When asserted, the core is enabled to process the pause-port frames. This signal is active when the **Rx Port/PFC** parameter is either set to **Port Enabled** or to **Port & PFC Enabled**.
I_CFG_SYS_PAUSE_RX_PFC_EN_W	I_SYS_CLK	8	Input	When asserted, the core is enabled to process the pause-pfc frames. This signal is active when the **Rx Port/PFC** parameter is either set to **Port Enabled** or to **Port & PFC Enabled**.
O_SYS_PAUSE_RX_PORT_XOFF	I_SYS_CLK	1	Output	The signal asserts when the core has received and processed a pause-port frame with an active pause time. This signal is active when the **Rx Port/PFC** parameter is either set to **Port Enabled** or to **Port & PFC Enabled**.
O_SYS_PAUSE_RX_PFC_XOFF_W	I_SYS_CLK	8	Output	The signal asserts when the core has received and processed a pause-pfc frame with an active pause time. O_SYS_PAUSE_RX_PFC_XOFF_W[0] is associated with priority 0, O_SYS_PAUSE_RX_PFC_XOFF_W[1] is associated with priority 1, and so on. This signal is active when the **Rx Port/PFC** parameter is either set to **Port Enabled** or to **Port & PFC Enabled**.
RS				
I_CFG_RS_TX_FAULT_EN	I_CORE_TX_CLK	1	Input	Configure Tx RS Fault enable. This signal enables the transmit core to respect the receive RS fault status. When asserted low, the core disregards the signals, and when asserted high, the core responds by transmitting the appropriate fault codes. For more information about behaviour, see **Reconciliation Sublayer**.
I_CFG_RS_TX_FAULT_LOCAL	I_CORE_TX_CLK	1	Input	The signal informs the RS transmit block to send local fault condition. Normally/By default, this signal is tied low.
I_CFG_RS_TX_FAULT_REMOTE	I_CORE_TX_CLK	1	Input	The signal informs the RS transmit block to send remote fault condition. Normally/By default, this signal is tied low.
I_CFG_RS_TX_IDLE	I_CORE_TX_CLK	1	Input	The signal informs the RS transmit block to continuously send idle code words. Normally/By default, this signal is tied low.

.....continued				
Name	**Clock Domain**	**Width**	**Direction**	**Description**
I_RS_RX_SRESET	I_CORE_RX_CLK	1	Input	Synchronous reset for receive RS logic. This signal is normally/by default connected to O_CORE_RX_SRESET.
O_RS_RX_FAULT_LOCAL	I_CORE_RX_CLK	1	Output	The signal indicates that the receive RS block is in local fault mode.
O_RS_RX_FAULT_REMOTE	I_CORE_RX_CLK	1	Output	The signal indicates that the receive RS block is in remote fault mode.
10GE TX PCS SIGNALS				
O_PMA49_TX_GRBX_SOS	I_CORE_TX_CLK	1	Output	Used when interfacing with SerDes with gearbox offload. Only available when 10GBASE-R is selected for the **10G Type** parameter.
O_PMA49_TX_GRBX_HDR_EN	I_CORE_TX_CLK	1	Output	Used when interfacing with SerDes with gearbox offload. Only available when 10GBASE-R is selected for the **10G Type** parameter.
O_PMA49_TX_GRBX_HDR	I_CORE_TX_CLK	4	Output	Used when interfacing with SerDes with gearbox offload. Only available when 10GBASE-R is selected for the **10G Type** parameter .
O_PMA49_TX_GRBX_DATA_EN	I_CORE_TX_CLK	1	Output	Used when interfacing with SerDes with gearbox offload. Only available when 10GBASE-R is selected for the **10G Type** parameter.
O_PMA49_TX_GRBX_DATA	I_CORE_TX_CLK	32 or 64	Output	Used when interfacing with SerDes with gearbox offload. Only available when 10GBASE-R is selected for the **10G Type** parameter .
I_PCS49_TX_SRESET	I_CORE_TX_CLK	1	Input	The transmit active high reset signal. Asserts for one or more clock cycles to reset the core.
I_CFG_PCS49_TX_BYPASS_SCRAMBLER	I_CORE_TX_CLK	1	Input	When asserted high, the scrambler is bypassed. This signal is normally/by default tied low.
I_CFG_PCS49_TX_TEST_PRBS31_EN	I_CORE_TX_CLK	1	Input	When asserted high, the core continuously sources PRBS31, as define by Clause49. This signal is the highest priority, that is if asserted, the value on the other I_CFG_PCS49_* signal is immaterial.
I_CFG_PCS49_TX_TEST_PATTERN_EN	I_CORE_TX_CLK	1	Input	When I_CFG_PCS49_TX_TEST_PATTERN_EN asserts, the transmitter sources either a square pattern or a pseudo-random pattern as defined by Clause49. The specific pattern is determined by I_CFG_PCS49_TX_TEST_PATTERN_TYPE_SEL.

.....continued				
Name	**Clock Domain**	**Width**	**Direction**	**Description**
I_CFG_PCS49_TX_TEST_PATTERN_TYPE_SEL	I_CORE_TX_CLK	1	Input	When asserted high, the core transmits a square pattern. The square pattern is sourced as 16'hf0f0 towards the PMA. When asserted low, the core sources a pseudo-random pattern as determined by I_CFG_PCS49_TX_TEST_PATTERN_DATA_SEL, I_CFG_PCS49_TX_TEST_PATTERN_SEED_A, and I_CFG_PCS49_TX_TEST_PATTERN_SEED_B.
I_CFG_PCS49_TX_TEST_PATTERN_DATA_SEL	I_CORE_TX_CLK	1	Input	When asserted high, the core uses 64 zeroes as the data-pattern. When asserted low, the core uses 64-bit encoding for the two Local Fault ordered sets. For clarification, see Clause 49.
I_CFG_PCS49_TX_TEST_PATTERN_SEED_A	I_CORE_TX_CLK	58	Input	These values are used for the scrambler seed while running in pseudo-random test-mode. For clarification, see Clause 49.
I_CFG_PCS49_TX_TEST_PATTERN_SEED_B	I_CORE_TX_CLK	58	Input	These values are used for the scrambler seed while running in pseudo-random test-mode. For clarification, see Clause 49.
10GE RX PCS SIGNALS				
I_PMA49_RX_GRBX_LOCK	I_CORE_RX_CLK	1	Input	Used when interfacing with SerDes with gearbox offload. Only available when 10GBASE-R is selected for the **10G Type** parameter.
I_PMA49_RX_GRBX_SOS	I_CORE_RX_CLK	1	Input	Used when interfacing with SerDes with gearbox offload. Only available when 10GBASE-R is selected for the **10G Type** parameter.
I_PMA49_RX_GRBX_HDR_EN	I_CORE_RX_CLK	1	Input	Used when interfacing with SerDes with gearbox offload. Only available when 10GBASE-R is selected for the **10G Type** parameter.
I_PMA49_RX_GRBX_HDR	I_CORE_RX_CLK	4	Input	Used when interfacing with SerDes with gearbox offload. Only available when 10GBASE-R is selected for the **10G Type** parameter.
I_PMA49_RX_GRBX_DATA_EN	I_CORE_RX_CLK	1	Input	Used when interfacing with SerDes with gearbox offload. Only available when 10GBASE-R is selected for the **10G Type** parameter.
I_PMA49_RX_GRBX_DATA	I_CORE_RX_CLK	32	Input	Used when interfacing with SerDes with gearbox offload. Only available when 10GBASE-R is selected for the **10G Type** parameter.

.....continued

Name	Clock Domain	Width	Direction	Description
I_PCS49_RX_SRESET	I_CORE_RX_CLK	1	Input	The receive active high reset signal. Asserts for one or more clock cycles to reset the core.
I_CFG_PCS49_RX_BYPASS_SCRAMBLER	I_CORE_RX_CLK	1	Input	When asserted high, the scrambler is bypassed. This signal is normally/by default tied low.
I_CFG_PCS49_RX_TEST_PRBS31_EN	I_CORE_RX_CLK	1	Input	When asserted high, the core continuously sources PRBS31, as define by Clause49. This signal is the highest priority, that is if asserted the value on the other I_CFG_PCS49_* signal is immaterial.
I_CFG_PCS49_RX_TEST_PATTERN_EN	I_CORE_RX_CLK	1	Input	When I_CFG_PCS49_RX_TEST_PATTERN_EN is asserted, the receiver sources either a square pattern or a pseudo-random pattern as defined by Clause49. The specific pattern is determined by I_CFG_PCS49_RX_TEST_PATTERN_TYPE_SEL.
I_CFG_PCS49_RX_TEST_PATTERN_TYPE_SEL	I_CORE_RX_CLK	1	Input	When asserted high, the core receives a square pattern. The square pattern is sourced as 16'hf0f0 towards the PMA. When asserted low, the core sources a pseudo-random pattern as determined by I_CFG_PCS49_RX_TEST_PATTERN_DATA_SEL.
I_CFG_PCS49_RX_TEST_PATTERN_DATA_SEL	I_CORE_RX_CLK	1	Input	When asserted high, the core uses 64 zeroes as the data-pattern. When asserted low, the core uses 64-bit encoding for the two Local Fault ordered sets. For clarification, see Clause 49.
10GE PCS RX STATUS & STATS				
O_PCS49_RX_BLOCK_LOCK	I_CORE_RX_CLK	1	Output	The signal asserts when the receiver acquires block delineation.
O_PCS49_RX_HI_BER	I_CORE_RX_CLK	1	Output	The signal asserts when the ber_cnt equals or exceeds 16 indicating a bit error ratio $>10^{-4}$.
O_PCS49_RX_STATUS	I_CORE_RX_CLK	1	Output	This signal indicates that the receiver is in block lock and not in hi_ber state.
O_PCS49_RX_BER_STRB	I_CORE_RX_CLK	1	Output	The signal strobes after every 125 micro seconds (us) as per the Clause49 specification. The O_PCS49_RX_BER_CNT signal is updated on the same event.

.....continued				
Name	**Clock Domain**	**Width**	**Direction**	**Description**
O_PCS49_RX_BER_CNT	I_CORE_RX_CLK	8	Output	An 8-bit counter that counts each time BER_BAD_SH state is entered. The counter reflects the number of events since the last time O_PCS49_RX_BER_STRB has asserted. The counter value is updated at the same time as O_PCS49_RX_BER_STRB and remains stable until the next O_PCS49_RX_BER_STRB event. The maximum value of the signal is 16.
O_PCS49_RX_TEST_MODE_ERR_STRB	I_CORE_RX_CLK	1	Output	The signal strobes to indicate an update on O_PCS49_RX_TEST_MODE_ERR_CNT. **Note**: This signal asserts irrespective of errors being present.
O_PCS49_RX_TEST_MODE_ERR_CNT	I_CORE_RX_CLK	8	Output	The receive test pattern error counter. This counter is used to indicate PRBS31 errors and pseudo-random-sequence errors. When the receiver is not running in one of those two modes, the counter is always zero, and the associated strobe never asserts.
O_PCS49_RX_ERRORED_BLOCK_CNT_STRB	I_CORE_RX_CLK	1	Output	When the receiver is in normal mode, this signal strobes each time RX_E state is entered.
APB INTERFACE				
PRESETN	PCLK	1	Input	Active low asynchronous reset. This reset must be synchronized externally in PCLK clock domain.
PWRITE	PCLK	1	Input	APB write/read enable, active high
PADDR	PCLK	16	Input	APB address
PSEL	PCLK	1	Input	APB select
PENABLE	PCLK	1	Input	APB enable
PWDATA	PCLK	8	Input	APB data input
PRDATA	PCLK	8	Output	APB data output
PREADY	PCLK	1	Output	Ready. The Slave uses this signal to extend an APB transfer.
PSLVERR	PCLK	1	Output	This signal indicates a transfer failure.
XGMII				
I_XGMII_RXD	I_CORE_RX_CLK	64	Input	XGMII Rx Control Only available when XGMII is selected for parameter **10G Type**
I_XGMII_RXC	I_CORE_RX_CLK	8	Input	XGMII Rx Control Only available when XGMII is selected for parameter **10G Type**

.....continued				
Name	**Clock Domain**	**Width**	**Direction**	**Description**
O_XGMII_TXD	I_CORE_TX_CLK	64	Output	XGMII Tx Control Only available when XGMII is selected for parameter **10G Type**
O_XGMII_TXC	I_CORE_TX_CLK	8	Output	XGMII Tx Control Only available when XGMII is selected for parameter **10G Type**

Note: Different combinations of these IO's are visible to user depending on parameter values.

4. Timing Diagrams

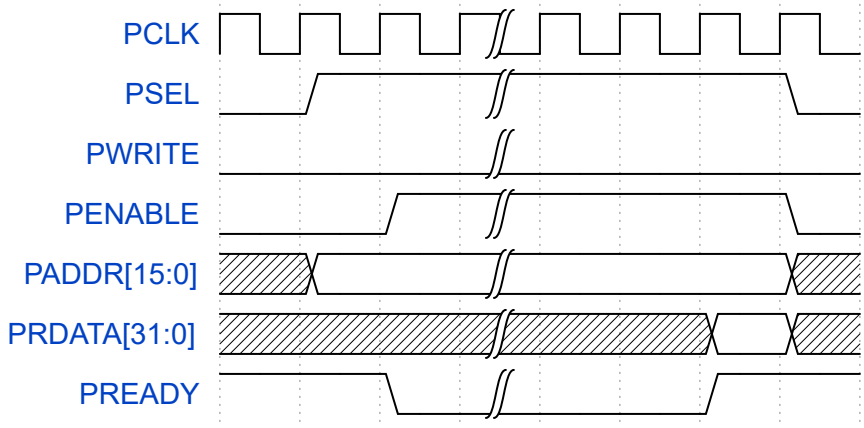
This section discusses various timing diagrams.

4.1 APB Interface

4.1.1 APB Read Timing

The following figure shows the timing diagram for an APB read access.

Figure 4-1. APB Read Timing Diagram

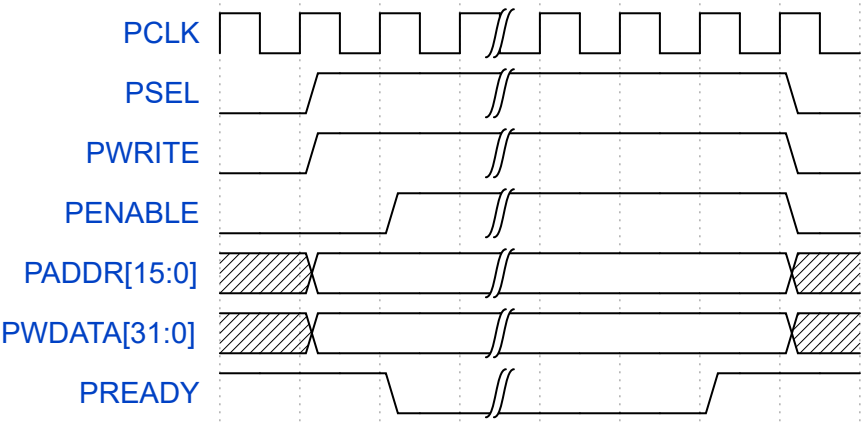


As per the preceding figure, the read transfer starts with PADDR, PWRITE, and PSEL changing after the rising edge of PCLK. The first clock cycle of the transfer is called the setup phase. After the following clock edge, the PENABLE is asserted and PREADY is de-asserted; this indicates that the access phase is taking place. PADDR, PWRITE, PSEL, and PENABLE all remain valid throughout the access phase. The transfer completes at the end of the cycle where PREADY is asserted. During this cycle, PRDATA is valid. PENABLE is de-asserted at the end of the transfer. PSEL also goes low unless the transfer is to be followed immediately by another transfer to the same peripheral.

4.1.2 APB Write Timing

The following figure shows the timing diagram for a APB write access.

Figure 4-2. APB Write Timing Diagram



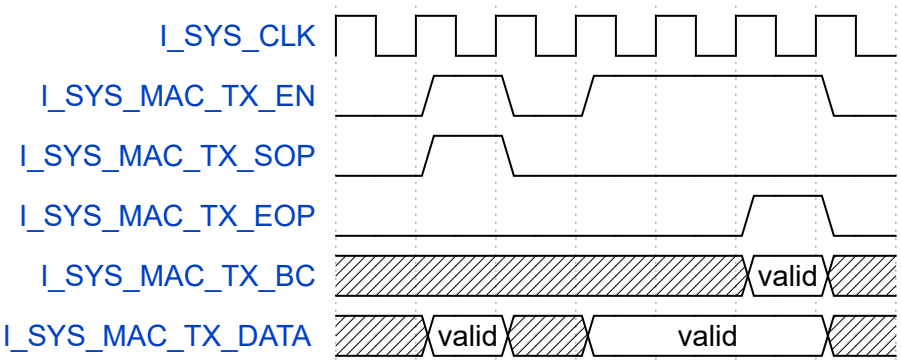
As per the preceding figure, the write transfer starts with PADDR, PWDATA, PWRITE, and PSEL changing after the rising edge of PCLK. The first clock cycle of the transfer is called the setup phase. After the following clock edge, the PENABLE is asserted and PREADY is de-asserted; this indicates that the access phase is taking place. PADDR, PWDATA, PWRITE, PSEL, and PENABLE all remain valid throughout the access phase. The transfer completes at the end of the cycle where PREADY is asserted. PENABLE is de-asserted at the end of the transfer. PSEL also goes low unless the transfer is to be followed immediately by another transfer to the same peripheral.

4.2 Dataplane

4.2.1 Tx Dataplane Basic Timing

The following figure shows the timing diagram for a packet transmit event.

Figure 4-3. Tx Dataplane Timing Diagram



The user initiates a packet transmit event by asserting I_SYS_MAC_TX_EN and I_SYS_MAC_TX_SOP. The packet transfer ends when the user asserts I_SYS_MAC_TX_EOP and I_SYS_MAC_TX_EN.

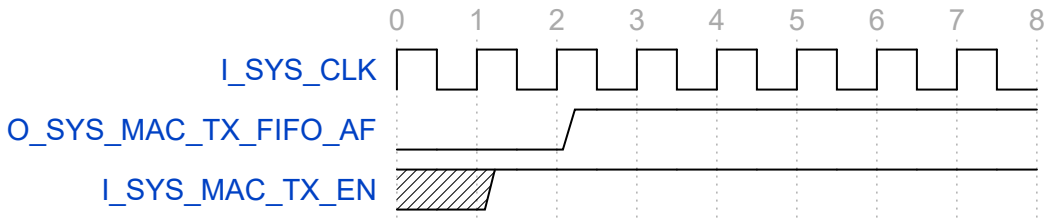
- I_SYS_MAC_TX_EN qualifies all signals
- I_SYS_MAC_TX_EOP qualifies I_SYS_MAC_TX_BC
- I_SYS_MAC_TX_EN assertion and deassertion can be performed at will, within the bounds of not causing a transmit FIFO underrun or overflow

All accesses are full accesses except the I_SYS_MAC_TX_EOP access, where I_SYS_MAC_TX_BC qualifies the number of active bytes.

Note: The receive interface signals are identical to the transmit interface signals.

The following figure shows the timing diagram for a transmit backpressure event.

Figure 4-4. Tx Dataplane Backpressure Timing Diagram

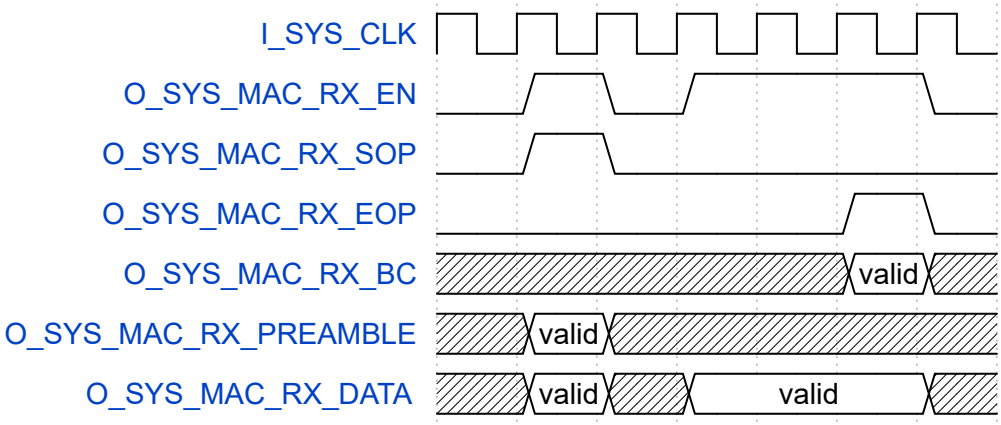


The diagram depicts a situation where the write on the rising edge of cycle 2 causes the almost full flag to assert. When this flag asserts, the system transmit FIFO is guaranteed to have space for four more writes, so the writes on cycle 3, 4, 5, and 6 are guaranteed to be written correctly, and the write on cycle 7 might be intelligently dropped.

4.2.2 Rx Dataplane Basic Timing

The following figure shows the timing diagram for a packet receive event.

Figure 4-5. Rx Dataplane Timing Diagram



A packet initiates when O_SYS_MAC_RX_EN and O_SYS_MAC_RX_SOP are asserted.

The packet transfer ends when O_SYS_MAC_RX_EOP and O_SYS_MAC_RX_EN are asserted.

- O_SYS_MAC_RX_EN qualifies all signals.
- O_SYS_MAC_RX_SOP qualifies O_SYS_MAC_RX_PREAMBLE.
- O_SYS_MAC_RX_EOP qualifies O_SYS_MAC_RX_BC.
- O_SYS_MAC_RX_EN assertion and deassertion can be performed at will, within the bounds of not causing a receive FIFO underrun or overflow.

All accesses are full accesses except the O_SYS_MAC_RX_EOP access, where O_SYS_MAC_RX_BC qualifies the number of active bytes.

4.3 Reset Controller

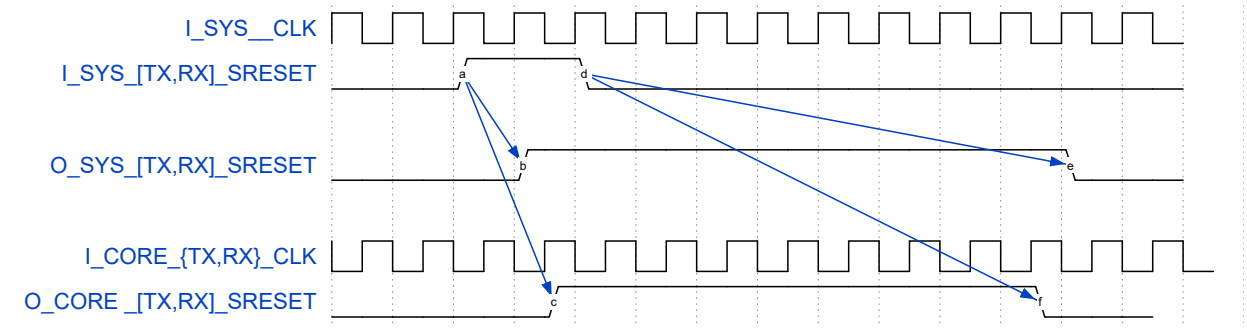
The transmit and receive MACs contain a reset controller. The simplest reset configuration is where the reset controller is used to reset everything. The master reset controller has the following signals:

- I_SYS_[TX,RX]_SRESET
- O_CORE_[TX,RX]_SRESET

As long as I_SYS_[TX,RX]_SRESET is asserted, O_CORE_[TX,RX]_SRESET are asserted. When I_SYS_[TX,RX]_SRESET is de-asserted, the core maintains assertion on O_CORE_[TX,RX]_SRESET, and once the required duration has passed, it releases O_CORE_[TX,RX]_SRESET in the correct order.

The following figure shows the MAC reset scheme for Tx/Rx.

Figure 4-6. Tx/Rx MAC Reset Timing Diagram



The duration of I_SYS_[TX,RX]_SRESET assertion can be from 1 to infinite clock cycles.

After I_SYS_[TX,RX]_SRESET is released, O_CORE_[TX,RX]_SRESET remain asserted for a predefined number of clock cycles, typically around 32–64 clock cycles.

I_SYS_TX_SRESET must be asserted until I_SYS_CLK and I_CORE_TX_CLK are stable.

I_SYS_RX_SRESET must be asserted until I_SYS_CLK and I_CORE_RX_CLK are stable.

5. Tool Flows

This section discusses various tool flow related information.

5.1 Licensing

Core10GMAC is available in two versions:

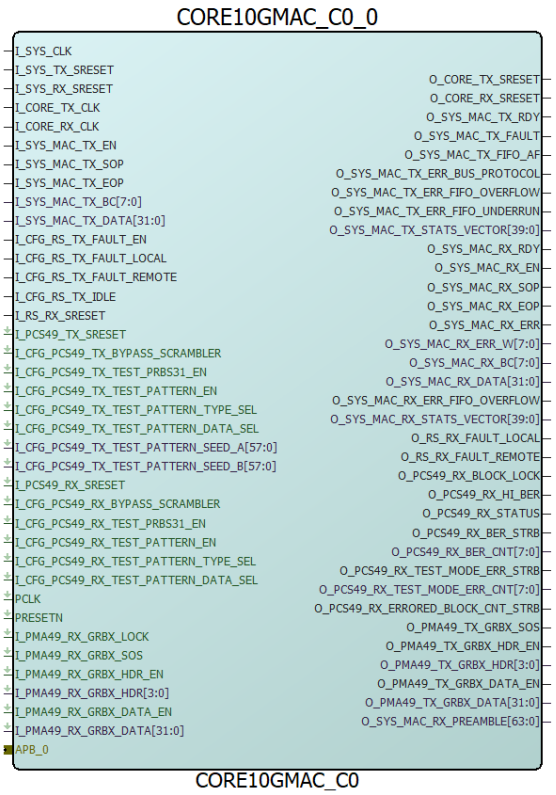
- **Obfuscated**: The obfuscated version is encrypted, license locked, and supports unlimited functionality on silicon. User needs to purchase this license separately.
- **Evaluation**: Evaluation version is available for free and supports four hours of the functionality on silicon .

5.2 SmartDesign

Core10GMAC is available for download to the SmartDesign IP catalog through the Libero SoC web repository. To know how to create SmartDesign project, see SmartDesign User Guide.

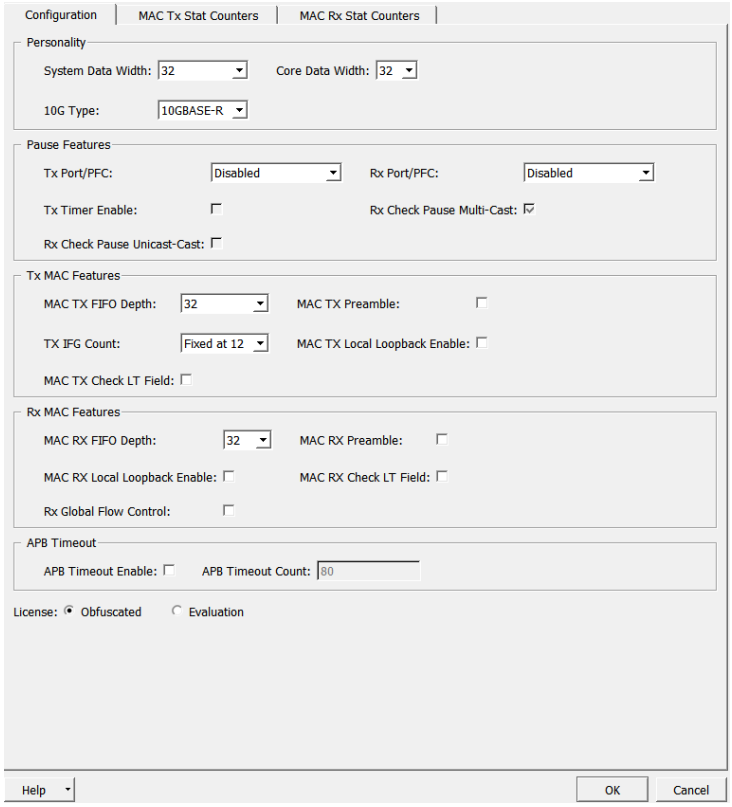
The following figure shows an example of an instantiated view of Core10GMAC on the SmartDesign canvas.

Figure 5-1. Core10GMAC Full I/O View



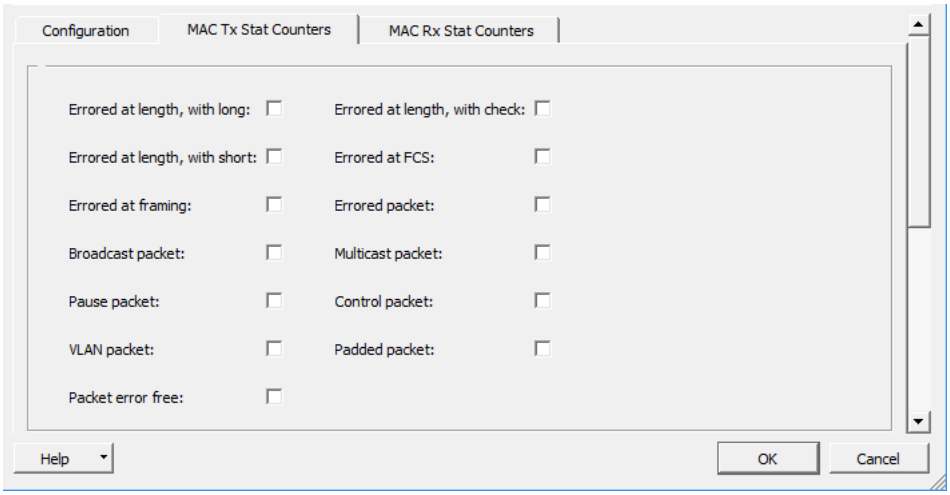
The following figure shows the options available under the **Configuration** tab.

Figure 5-2. Core10GMAC SmartDesign Configuration GUI – Configuration



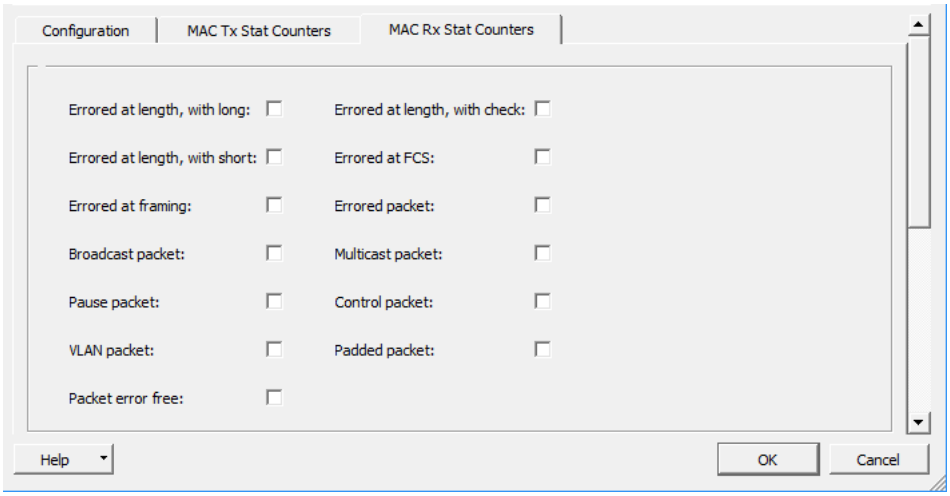
The following figure shows the options available under the **MAC Tx Stat Counters** tab.

Figure 5-3. Core10GMAC SmartDesign Configuration GUI – MAC Tx Stat Counters



The following figure shows the options available under the **MAC Rx Stat Counters**tab.

Figure 5-4. Core10GMAC SmartDesign Configuration GUI – MAC Rx Stat Counters



5.3 Simulation Flows

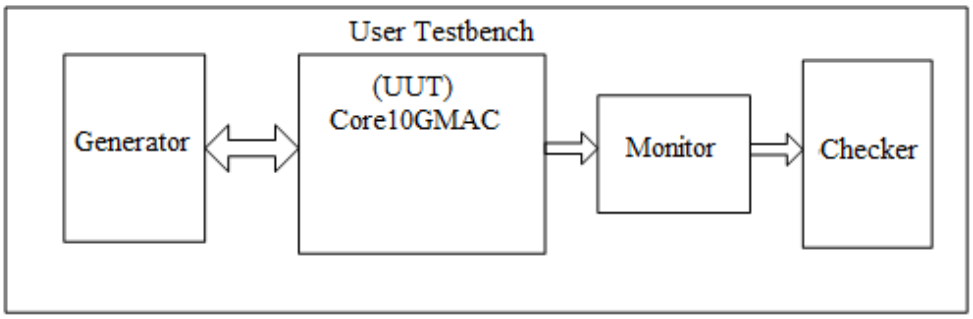
SmartDesign and Libero SoC facilitate running a user testbench for Core10GMAC. To run the user testbench, perform the following steps to set the design root to be the Core10GMAC instance.

1. Click the **Stimulus Hierarchy** tab.
2. Right-click testbench, under **Simulate Pre-Synthesis Design**, select **Open Interactively**. Separate Modelsim window opens; wait for Modelsim to complete the simulation and display results on the **Transcript** tab in Modelsim tool.

5.3.1 User Testbench

The Core10GMAC user testbench gives an example of how to use the core.

Figure 5-5. Core10GMAC User Testbench



The simulation testbench in the preceding figure includes an instantiation of the Core10GMAC macro, data generation, and data monitor and checker. The purpose of the testbench is to test the functionality of the core by providing known data, monitoring the output, and checking for expected results.

The core is delivered with a simple simulation test-bench. This test-bench is purely delivered as a vehicle to get started with using the core, and it is not an attempt at an exhaustive testbench.

5.4 Timing Constraint

To meet the Core10GMAC timing requirement, it is important to provide timing constraint. You must provide set clock group timing constraint to set false path between asynchronous clocks. Core10GMAC uses the following clocks

- I_SYS_CLK
- I_CORE_TX_CLK
- I_CORE_RX_CLK
- PCLK

For example, if all the four clocks mentioned above are asynchronous then you can provide timing constraint as the following example:

```
set_clock_group -asynchronous -group [get_clocks {I_SYS_CLK}] \
                -group [get_clocks {I_CORE_TX_CLK}] \
                -group [get_clocks {I_CORE_RX_CLK}] \
                -group [get_clocks {PCLK}]
```

If I_SYS_CLK and I_CORE_TX_CLK are connected to same clock source then user can provide timing constraint as the following example:

```
set_clock_group -asynchronous -group [get_clocks {I_CORE_TX_CLK}] \
                              -group [get_clocks {I_CORE_RX_CLK}] \
                              -group [get_clocks {PCLK}]
```

The source clock name used in the above examples are for reference only. The source name may be different in your design and you must change the clock name in the constraint accordingly.

You can find reference sdc timing constraint file from the following path.

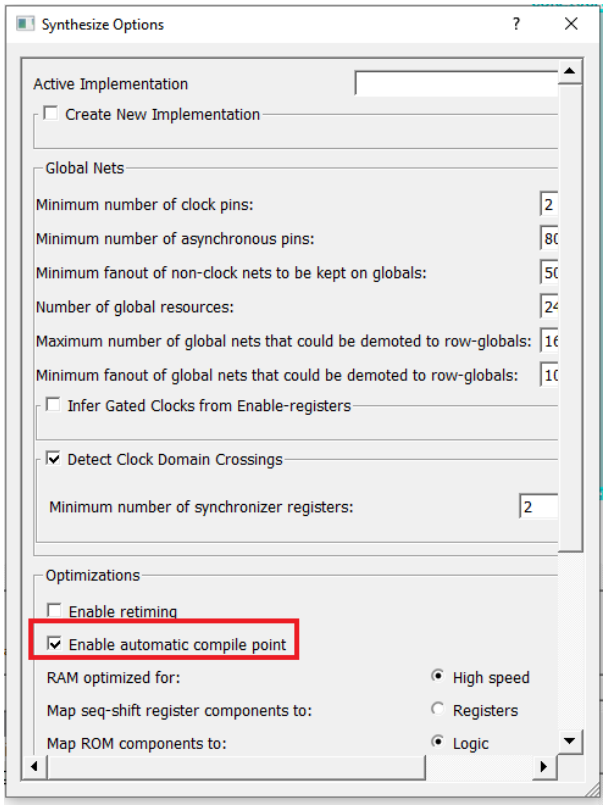
```
<project_directory/component/Actel/DirectCore/Core10GMAC/(example 3.0.xxx)/
constraints/Core10GMAC.sdc>
```

5.5 Synthesis in Libero SoC

To run synthesis with the configuration selected in the configuration GUI, set the design root appropriately. To meet the Core10GMAC timing requirements, the **Enable automatic compile point** synthesis constraint must be applied. To apply synthesis constraint, performs the following steps.

1. In the **Design Flow tab**, under **Implement Design**, right-click **Synthesize** and select **Configure Options**.
2. In the **Synthesis Options** window , select **Enable automatic compile point** option as shown in the following figure.
3. Click **OK**.

Figure 5-6. Synthesis Constraint



4. After setting the synthesis constraint, to run the synthesis, double click on **Synthesize**.

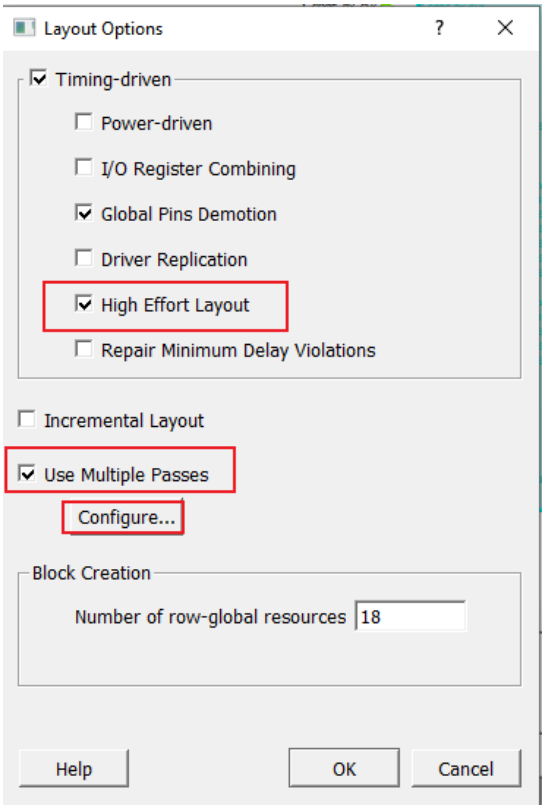
Note: Synthesis constraint is must when the **Core Data Width** parameter is set to 32.

5.6 Place-and-Route in Libero SoC

To meet the Core10GMAC timing requirement, place and route constraint must be applied. To apply place and route constraint, perform the following steps.

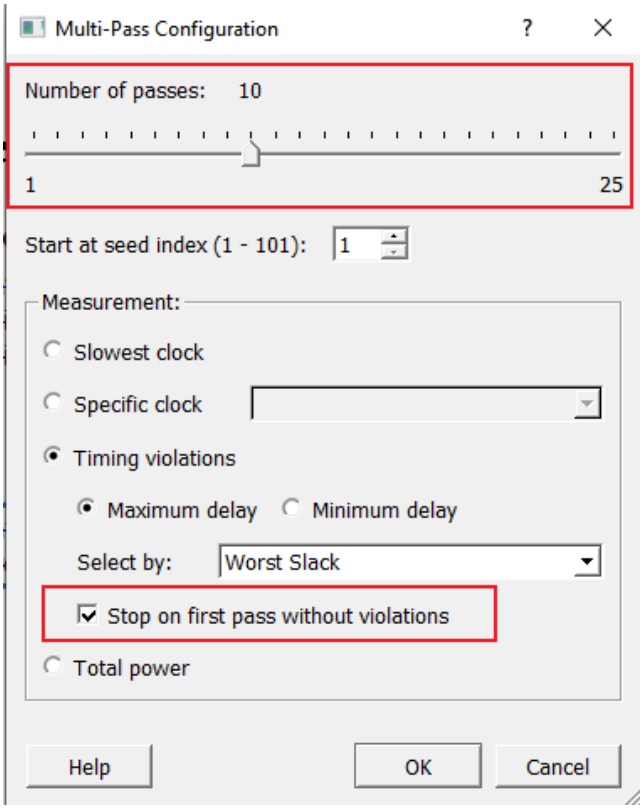
1. In the **Design Flow** tab, under **Implement Design**, right-click **Place and Route**, and then select **Configure Options**.
2. In the **Layout Options** window, select **High Effort Layout** and **Use Multiple Passes** option. After selecting **Use Multiple Passes**, click **Configure** as shown in the following figure.

Figure 5-7. Place and Route Constraints



3. In the **Multi-Pass configuration** window, set **Number of passes** to 10, and then select **Stop on first pass without violations** option as shown in the following figure.

Figure 5-8. Place and Route Constraints Continue..



4. Click **OK**.
5. After applying constraint, to run the place and route, double-click **Place and Route.**

Note: Place and Route constraint is must when the **Core Data Width** parameter is set to 32.

6. Revision History

The revision history describes the changes that were implemented in the document. The changes are listed by revision, starting with the most current publication.

Table 6-1. Revision History

Revision	Date	Description
A	05/2022	The following is the list of changes in revision A of the document: • The document was migrated to the Microchip template. • The document number was updated to DS50003329A from 50200765.
4.0	—	Added PolarFire SoC support.
3.0	—	Updated for Core10GMAC v2.2. Added two parameters "APB_TIMEOUT_EN" and "APB_TIMEOUT_COUNT".
2.0	—	Updated for Core10GMAC v2.1.
1.0	—	Revision 1.0 was the first publication of this document. Created for Core10GMAC v2.0.

Microchip FPGA Support

Microchip FPGA products group backs its products with various support services, including Customer Service, Customer Technical Support Center, a website, and worldwide sales offices. Customers are suggested to visit Microchip online resources prior to contacting support as it is very likely that their queries have been already answered.

Contact Technical Support Center through the website at www.microchip.com/support. Mention the FPGA Device Part number, select appropriate case category, and upload design files while creating a technical support case.

Contact Customer Service for non-technical product support, such as product pricing, product upgrades, update information, order status, and authorization.

- From North America, call **800.262.1060**
- From the rest of the world, call **650.318.4460**
- Fax, from anywhere in the world, **650.318.8044**

The Microchip Website

Microchip provides online support via our website at www.microchip.com/. This website is used to make files and information easily available to customers. Some of the content available includes:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQs), technical support requests, online discussion groups, Microchip design partner program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

Product Change Notification Service

Microchip's product change notification service helps keep customers current on Microchip products. Subscribers will receive email notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, go to www.microchip.com/pcn and follow the registration instructions.

Customer Support

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Embedded Solutions Engineer (ESE)
- Technical Support

Customers should contact their distributor, representative or ESE for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in this document.

Technical support is available through the website at: www.microchip.com/support

Microchip Devices Code Protection Feature

Note the following details of the code protection feature on Microchip products:

- Microchip products meet the specifications contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is secure when used in the intended manner, within operating specifications, and under normal conditions.

- Microchip values and aggressively protects its intellectual property rights. Attempts to breach the code protection features of Microchip product is strictly prohibited and may violate the Digital Millennium Copyright Act.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of its code. Code protection does not mean that we are guaranteeing the product is “unbreakable”. Code protection is constantly evolving. Microchip is committed to continuously improving the code protection features of our products.

Legal Notice

This publication and the information herein may be used only with Microchip products, including to design, test, and integrate Microchip products with your application. Use of this information in any other manner violates these terms. Information regarding device applications is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. Contact your local Microchip sales office for additional support or, obtain additional support at www.microchip.com/en-us/support/design-help/client-support-services.

THIS INFORMATION IS PROVIDED BY MICROCHIP "AS IS". MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE, OR WARRANTIES RELATED TO ITS CONDITION, QUALITY, OR PERFORMANCE.

IN NO EVENT WILL MICROCHIP BE LIABLE FOR ANY INDIRECT, SPECIAL, PUNITIVE, INCIDENTAL, OR CONSEQUENTIAL LOSS, DAMAGE, COST, OR EXPENSE OF ANY KIND WHATSOEVER RELATED TO THE INFORMATION OR ITS USE, HOWEVER CAUSED, EVEN IF MICROCHIP HAS BEEN ADVISED OF THE POSSIBILITY OR THE DAMAGES ARE FORESEEABLE. TO THE FULLEST EXTENT ALLOWED BY LAW, MICROCHIP'S TOTAL LIABILITY ON ALL CLAIMS IN ANY WAY RELATED TO THE INFORMATION OR ITS USE WILL NOT EXCEED THE AMOUNT OF FEES, IF ANY, THAT YOU HAVE PAID DIRECTLY TO MICROCHIP FOR THE INFORMATION.

Use of Microchip devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights unless otherwise stated.

Trademarks

The Microchip name and logo, the Microchip logo, Adaptec, AnyRate, AVR, AVR logo, AVR Freaks, BesTime, BitCloud, CryptoMemory, CryptoRF, dsPIC, flexPWR, HELDO, IGLOO, JukeBlox, KeeLoq, Kleer, LANCheck, LinkMD, maXStylus, maXTouch, MediaLB, megaAVR, Microsemi, Microsemi logo, MOST, MOST logo, MPLAB, OptoLyzer, PIC, picoPower, PICSTART, PIC32 logo, PolarFire, Prochip Designer, QTouch, SAM-BA, SenGenuity, SpyNIC, SST, SST Logo, SuperFlash, Symmetricom, SyncServer, Tachyon, TimeSource, tinyAVR, UNI/O, Vectron, and XMEGA are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

AgileSwitch, APT, ClockWorks, The Embedded Control Solutions Company, EtherSynch, Flashtec, Hyper Speed Control, HyperLight Load, IntelliMOS, Libero, motorBench, mTouch, Powermite 3, Precision Edge, ProASIC, ProASIC Plus, ProASIC Plus logo, Quiet- Wire, SmartFusion, SyncWorld, Temux, TimeCesium, TimeHub, TimePictra, TimeProvider, TrueTime, WinPath, and ZL are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Adjacent Key Suppression, AKS, Analog-for-the-Digital Age, Any Capacitor, AnyIn, AnyOut, Augmented Switching, BlueSky, BodyCom, CodeGuard, CryptoAuthentication, CryptoAutomotive, CryptoCompanion, CryptoController, dsPICDEM, dsPICDEM.net, Dynamic Average Matching, DAM, ECAN, Espresso T1S, EtherGREEN, GridTime, IdealBridge, In-Circuit Serial Programming, ICSP, INICnet, Intelligent Paralleling, Inter-Chip Connectivity, JitterBlocker, Knob-on-Display, maxCrypto, maxView, memBrain, Mindi, MiWi, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, MultiTRAK, NetDetach, NVM Express, NVMe, Omniscient Code Generation, PICDEM, PICDEM.net, PICkit, PICtail, PowerSmart, PureSilicon, QMatrix, REAL ICE, Ripple Blocker, RTAX, RTG4, SAM-ICE, Serial Quad I/O, simpleMAP, SimpliPHY, SmartBuffer, SmartHLS, SMART-I.S., storClad, SQI, SuperSwitcher, SuperSwitcher II, Switchtec, SynchroPHY, Total Endurance, TSHARC, USBCheck, VariSense, VectorBlox, VeriPHY,

ISBN: 978-1-6683-0428-0

Quality Management System

For information regarding Microchip's Quality Management Systems, please visit www.microchip.com/quality.

Worldwide Sales and Service

AMERICAS	ASIA/PACIFIC	ASIA/PACIFIC	EUROPE
Corporate Office 2355 West Chandler Blvd. Chandler, AZ 85224-6199 Tel: 480-792-7200 Fax: 480-792-7277 Technical Support: www.microchip.com/support Web Address: www.microchip.com **Atlanta** Duluth, GA Tel: 678-957-9614 Fax: 678-957-1455 **Austin, TX** Tel: 512-257-3370 **Boston** Westborough, MA Tel: 774-760-0087 Fax: 774-760-0088 **Chicago** Itasca, IL Tel: 630-285-0071 Fax: 630-285-0075 **Dallas** Addison, TX Tel: 972-818-7423 Fax: 972-818-2924 **Detroit** Novi, MI Tel: 248-848-4000 **Houston, TX** Tel: 281-894-5983 **Indianapolis** Noblesville, IN Tel: 317-773-8323 Fax: 317-773-5453 Tel: 317-536-2380 **Los Angeles** Mission Viejo, CA Tel: 949-462-9523 Fax: 949-462-9608 Tel: 951-273-7800 **Raleigh, NC** Tel: 919-844-7510 **New York, NY** Tel: 631-435-6000 **San Jose, CA** Tel: 408-735-9110 Tel: 408-436-4270 **Canada - Toronto** Tel: 905-695-1980 Fax: 905-695-2078	**Australia - Sydney** Tel: 61-2-9868-6733 **China - Beijing** Tel: 86-10-8569-7000 **China - Chengdu** Tel: 86-28-8665-5511 **China - Chongqing** Tel: 86-23-8980-9588 **China - Dongguan** Tel: 86-769-8702-9880 **China - Guangzhou** Tel: 86-20-8755-8029 **China - Hangzhou** Tel: 86-571-8792-8115 **China - Hong Kong SAR** Tel: 852-2943-5100 **China - Nanjing** Tel: 86-25-8473-2460 **China - Qingdao** Tel: 86-532-8502-7355 **China - Shanghai** Tel: 86-21-3326-8000 **China - Shenyang** Tel: 86-24-2334-2829 **China - Shenzhen** Tel: 86-755-8864-2200 **China - Suzhou** Tel: 86-186-6233-1526 **China - Wuhan** Tel: 86-27-5980-5300 **China - Xian** Tel: 86-29-8833-7252 **China - Xiamen** Tel: 86-592-2388138 **China - Zhuhai** Tel: 86-756-3210040	**India - Bangalore** Tel: 91-80-3090-4444 **India - New Delhi** Tel: 91-11-4160-8631 **India - Pune** Tel: 91-20-4121-0141 **Japan - Osaka** Tel: 81-6-6152-7160 **Japan - Tokyo** Tel: 81-3-6880- 3770 **Korea - Daegu** Tel: 82-53-744-4301 **Korea - Seoul** Tel: 82-2-554-7200 **Malaysia - Kuala Lumpur** Tel: 60-3-7651-7906 **Malaysia - Penang** Tel: 60-4-227-8870 **Philippines - Manila** Tel: 63-2-634-9065 **Singapore** Tel: 65-6334-8870 **Taiwan - Hsin Chu** Tel: 886-3-577-8366 **Taiwan - Kaohsiung** Tel: 886-7-213-7830 **Taiwan - Taipei** Tel: 886-2-2508-8600 **Thailand - Bangkok** Tel: 66-2-694-1351 **Vietnam - Ho Chi Minh** Tel: 84-28-5448-2100	**Austria - Wels** Tel: 43-7242-2244-39 Fax: 43-7242-2244-393 **Denmark - Copenhagen** Tel: 45-4485-5910 Fax: 45-4485-2829 **Finland - Espoo** Tel: 358-9-4520-820 **France - Paris** Tel: 33-1-69-53-63-20 Fax: 33-1-69-30-90-79 **Germany - Garching** Tel: 49-8931-9700 **Germany - Haan** Tel: 49-2129-3766400 **Germany - Heilbronn** Tel: 49-7131-72400 **Germany - Karlsruhe** Tel: 49-721-625370 **Germany - Munich** Tel: 49-89-627-144-0 Fax: 49-89-627-144-44 **Germany - Rosenheim** Tel: 49-8031-354-560 **Israel - Ra'anana** Tel: 972-9-744-7705 **Italy - Milan** Tel: 39-0331-742611 Fax: 39-0331-466781 **Italy - Padova** Tel: 39-049-7625286 **Netherlands - Drunen** Tel: 31-416-690399 Fax: 31-416-690340 **Norway - Trondheim** Tel: 47-72884388 **Poland - Warsaw** Tel: 48-22-3325737 **Romania - Bucharest** Tel: 40-21-407-87-50 **Spain - Madrid** Tel: 34-91-708-08-90 Fax: 34-91-708-08-91 **Sweden - Gothenberg** Tel: 46-31-704-60-40 **Sweden - Stockholm** Tel: 46-8-5090-4654 **UK - Wokingham** Tel: 44-118-921-5800 Fax: 44-118-921-5820