HB0765
Handbook
Core10GMAC Version 2.2

Microsemi
a Microchip company

Microsemi Corporate Headquarters
One Enterprise, Aliso Viejo,
CA 92656 USA
Within the USA: +1 (800) 713-4113
Outside the USA: +1 (949) 380-6100
Sales: +1 (949) 380-6136
Fax: +1 (949) 215-4996
E-mail: sales.support@microsemi.com
www.microsemi.com

About Microsemi

Microsemi Corporation (Nasdaq: MSCC) offers a comprehensive portfolio of semiconductor and system solutions for aerospace & defense, communications, data center and industrial markets. Products include high-performance and radiation-hardened analog mixed-signal integrated circuits, FPGAs, SoCs and ASICs; power management products; timing and synchronization devices and precise time solutions, setting the world's standard for time; voice processing devices; RF solutions; discrete components; enterprise storage and communication solutions, security technologies and scalable anti-tamper products; Ethernet solutions; Power-over-Ethernet ICs and midspans; as well as custom design capabilities and services. Microsemi is headquartered in Aliso Viejo, Calif., and has approximately 4,800 employees globally. Learn more at www.microsemi.com.

50200765. 3.0 12/18

Contents

1 Revision History 1
- 1.1 Revision 3.0 1
- 1.2 Revision 2.0 1
- 1.3 Revision 1.0 1

2 Introduction 2
- 2.1 Key Features 2
- 2.2 Core Version 2
- 2.3 Supported Families 2
- 2.4 Utilization and Performance 2

3 Functional Descriptions 4
- 3.1 Core10GMAC Blocks 5
  - 3.1.1 Ethernet MAC / RS / PAUSE 5
  - 3.1.2 Physical Coding Sub-layer 5
  - 3.1.3 Link Training 5
  - 3.1.4 Auto-Negotiation 6

4 Operation 7
- 4.1 APB Control Registers 7
- 4.2 Address Map 7
- 4.3 Nomenclature 19
  - 4.3.1 Static Configuration Registers 19
  - 4.3.2 Dynamic Configuration Signals 19
- 4.4 Ethernet MAC/RS Overview 19
  - 4.4.1 Transmit FIFO 19
  - 4.4.2 Receive FIFO 19
  - 4.4.3 MAC Tx 20
  - 4.4.4 MAC Rx 20
  - 4.4.5 Statistics 20
- 4.5 Ethernet Interface 20
  - 4.5.1 Tx Dataplane 20
  - 4.5.2 Rx Dataplane 21
  - 4.5.3 System Interface Remote Loopback 22
  - 4.5.4 Local Loopback 23
  - 4.5.5 Pause Interface 23
  - 4.5.6 Clocking & Resets 25
  - 4.5.7 Initialization 25
  - 4.5.8 Statistics Vector 26
  - 4.5.9 Reconciliation Sublayer 27

5 Interface Description 28
- 5.1 Configuration GUI Parameters 28
- 5.2 I/O Signals 32

6 Timing Diagrams 45
- 6.1 APB Interface 45
  - 6.1.1 APB Read Timing 45
  - 6.1.2 APB Write Timing 45
- 6.2 Dataplane 46

6.2.1 Tx Dataplane Basic Timing 46
6.2.2 Rx Dataplane Basic Timing 47
6.3 Master Reset 47

7 Tool Flows 49
7.1 Licensing 49
7.1.1 Obfuscated 49
7.1.2 Evaluation 49
7.2 SmartDesign 50
7.3 Simulation Flows 53
7.3.1 User Testbench 53
7.4 Synthesis in Libero SoC PolarFire 53
7.5 Place-and-Route in Libero SoC PolarFire 53
7.6 Constraints 53
7.6.1 Synthesis Constraints 53

8 Ordering Information 54
8.1 Ordering Codes 54

Figures

Figure 1 10GBASE-R System Level Diagram . . . 4
Figure 2 10GBASE-KR System Level Diagram . . . 4
Figure 3 XGMII System Level Diagram . . . 5
Figure 4 Ethernet TX/Rx FIFO Block Diagram . . . 19
Figure 5 Pause Tx Example Configuration . . . 24
Figure 6 Pause Tx Example Timing Diagram . . . 24
Figure 7 FPGA Clocking Architecture . . . 25
Figure 8 APB Read Timing Diagram . . . 45
Figure 9 APB Write Timing Diagram . . . 45
Figure 10 Tx Dataplane Timing Diagram . . . 46
Figure 11 Tx Dataplane Backpressure Timing Diagram . . . 46
Figure 12 Rx Dataplane Timing Diagram . . . 47
Figure 13 Tx/Rx MAC Reset Timing Diagram . . . 47
Figure 14 Core10GMAC Full I/O View . . . 50
Figure 15 Core10GMAC SmartDesign Configuration GUI – Configuration . . . 51
Figure 16 Core10GMAC SmartDesign Configuration GUI – MAC Tx Stat Counters . . . 52
Figure 17 Core10GMAC SmartDesign Configuration GUI – MAC Rx Stat Counters . . . 52
Figure 18 Core10GMAC User Testbench . . . 53

Tables

Table 1 Core10GMAC Utilization for 10GBASE-R Design 2
Table 2 Core10GMAC Utilization for 10GBASE-KR Design 2
Table 3 Core10GMAC Utilization for XGMII Design 3
Table 4 Address Map 7
Table 5 Auto-Negotiation Tx Register 7
Table 6 Auto-Negotiation Rx Register 8
Table 7 Link Training Tx Register 10
Table 8 Link Training Rx Register 11
Table 9 Tx Ctrl Register 11
Table 10 Rx Ctrl Register 12
Table 11 MAC Tx Config Register 13
Table 12 MAC Rx Config Register 16
Table 13 MAC Tx Static Register 17
Table 14 MAC Rx Static Register 18
Table 15 Tx Dataplane Bus Protocol Encoding 20
Table 16 Encoding of I_SYS_MAC_TX_BC 21
Table 17 Rx Dataplane Bus Protocol Encoding 21
Table 18 Encoding of O_SYS_MAC_RX_BC 22
Table 19 Encoding of the Vector 26
Table 20 Priority of the Fault State Generation 27
Table 21 GUI Parameters 28
Table 22 Port Signals for Core10GMAC Macro 32
Table 23 Ordering Codes 54

1 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.

1.1 Revision 3.0

Updated for Core10GMAC v2.2. Added two parameters "APB_TIMEOUT_EN" and "APB_TIMEOUT_COUNT". Refer to Table 21, page 28.

1.2 Revision 2.0

Updated for Core10GMAC v2.1.

1.3 Revision 1.0

Revision 1.0 was the first publication of this document. Created for Core10GMAC v2.0.

2 Introduction

The Core10GMAC is designed for the IEEE 802.3-2012 specification and supports 10GBASE-R, 10GBASE-KR, and XGMII interfaces. This configurable core provides the complete MAC and PHY layer when used with a transceiver interface. The physical layer is designed to work seamlessly with the PolarFire transceiver using either the PMA or 64b/66b interface modes.

This handbook provides information on the Core10GMAC and the features it supports. This IP is part of the 10GbE subsystem, which is defined in the *UG0727: PolarFire FPGA 10G Ethernet Solutions User Guide*. This document provides information on how 10GbE can be implemented in PolarFire devices. For more information on the PolarFire transceivers, refer to *UG0677: PolarFire FPGA Transceiver User Guide*.

2.1 Key Features

The following are the key features of Core10GMAC:

- Ethernet MAC / RS / PAUSE
- Physical Coding Sub-layer
- Link Training
- Auto-Negotiation

2.2 Core Version

This handbook is for Core10GMAC version 2.2.

2.3 Supported Families

PolarFire®

2.4 Utilization and Performance

Core10GMAC has been implemented in the following Microsemi device families. A summary of the implementation data for Core10GMAC configured for 10GBASE-R is listed in Table 1.

Table 1 • **Core10GMAC Utilization for 10GBASE-R Design**

	Logic Elements		Utilization		Performance
Family	**Sequential**	**Combinatorial**	**Device**	**Total %**	**MHz**
PolarFire	5209	5490	MPF300TS	4.1	323

Note: 10G-BASE-R design connects to Microsemi's SerDes through the Gearbox Interface.

A summary of the implementation data for Core10GMAC configured for 10GBASE-KR is listed in Table 2.

Table 2 • **Core10GMAC Utilization for 10GBASE-KR Design**

	Logic Elements		Utilization		Performance
Family	**Sequential**	**Combinatorial**	**Device**	**Total %**	**MHz**
PolarFire	6196	7001	MPF300TS	4.3	323

Note: 10G-BASE-KR design connects to Microsemi's SerDes through the PMA Interface.

A summary of the implementation data for Core10GMAC configured for XGMII is listed in Table 3.

Table 3 • Core10GMAC Utilization for XGMII Design

Family	Logic Elements		Utilization		Performance MHz
	Sequential	Combinatorial	Device	Total %	
PolarFire	3561	3531	MPF300TS	2.37	323

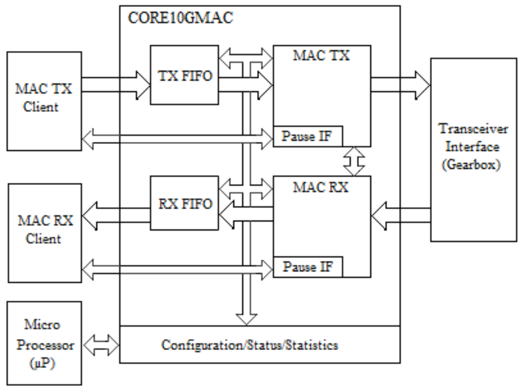
Note: XGMII design connects to Microsemi's SerDes through the PCS interface using an XGMII to PCS IP (for example, CoreXAUI).

3 Functional Descriptions

When Core10GMAC is configured for 10GBASE-R design the Transceiver 64b/66b Interface is used, and it connects directly to the MAC. Supports 32-bit or 64-bit datapath configuration. 10GBASE-R consists one main block: MAC. The MAC block supports Ethernet MAC RS/PAUSE.

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

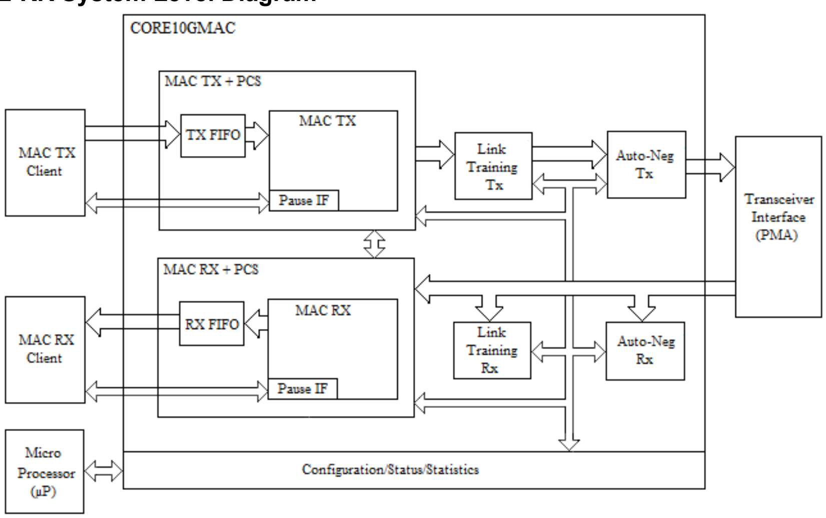
Figure 1 • **10GBASE-R System Level Diagram**



When Core10GMAC is configured for 10GBASE-KR design, the Transceiver PMA Interface is used and it connects to the Ethernet MAC through the PCS Interface. Supports 32-bit datapath configuration. For 10GBASE-KR configuration, the Link Training and Auto-Negotion Tx/Rx blocks are enable and they can be accessed from the 32-bit APB slave Interface. 10GBASE-KR consists four main blocks: MAC PCS, Link Training, and Auto-Negotion. The MAC PCS block supports Ethernet MAC RS/PAUSE.

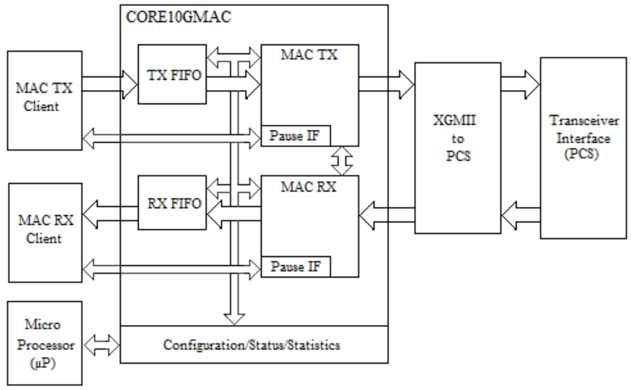
The following figure shows the Core 10GBASE-KR system level block diagram.

Figure 2 • **10GBASE-KR System Level Diagram**



When Core10GMAC is configured for XGMII design the PCS Interface is used, and it connects to the MAC through an XGMII to PCS IP (for example, CoreXAUI). Supports 64-bit datapath configuration. XGMII design consists one main block: MAC. The MAC block supports Ethernet MAC RS/PAUSE.

Figure 3 • **XGMII System Level Diagram**



3.1 Core10GMAC Blocks

The Core10GMAC contains the following blocks.

3.1.1 Ethernet MAC / RS / PAUSE

The MAC supports the following features:

- Configurable System Interface Bus Width, which supports 32-bit or 64-bit.
- 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 2 words for 10GE.
- Pause Frame Insertion on Transmit, and Flagging on receive
- Ethernet statistics on transmit and receive

3.1.2 Physical Coding Sub-layer

The PCS block supports the following features:

- Compliant with IEEE802.3 Clause 49, that is, PCS Sublayer for 64B/66B.
- 32bit or 64bit datapath connection to the Transceiver Interface.

Note: Support with IEEE802.3 Clause 36 to be added, that is, PCS Sublayer for 8B/10B.

3.1.3 Link Training

The Link Training block is compliant with IEEE 802.3-Clause 72 and supports the following features:

- **Transmit State-machine:** Controls the transmission of the training frame, which consists of the frame marker, coefficient update, status report and training pattern. The link training procedure is driven using a provided firmware driver, which can be run on a local processor and accessed through the APB interface.
- **Receive State-machine:** Controls the reception of the training frames by hunting for frame markers, performing bit slip and synchronizing to the frames. The received coefficient update and status report is handled by the firmware driver over the APB interface. The firmware driver will use this information to update transmitter emphasis as instructed by the Link partner.

3.1.4 Auto-Negotiation

The Auto-Negotiation block supports the ability to determine if the link is 10G KR, KX, or KX4. It is compliant with IEEE 802.3-Clause 73 and supports the following features:

- Transmit State-machine
- Receive State-machine
- Arbitration State-machine
- Next Page
- 32-bit APB slave interface to initialize and read results of negotiation.
- Default autonomous operation

A provided firmware driver handles the auto-negotiation process and the initialization and exchange of configuration pages.

4 Operation

4.1 APB Control Registers

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

4.2 Address Map

The following table lists the definition of PADDR[9:6] decoding and the explanation of the APB registers.

Table 4 • **Address Map**

Address	Name
0x0	Auto-Negotiation Tx Register
0x1	Auto-Negotiation Rx Register
0x2	Link Training Tx Register
0x3	Link Training Rx Register
0x8	Tx Ctlr Register
0x9	Rx Ctlr Register
0xA	MAC Tx Config Register
0xB	MAC Rx Config Register
0xC	MAC Tx Static Register
0xD	MAC Rx Static Register

The following tables lists the APB registers functionality. The offset column represents PADDR[5:2].

Table 5 • **Auto-Negotiation Tx Register**

Offset	Register Name	Bit(s) Name	Bit	Default	Action	Description
0	Main Control	Reset	7	0	RW	Dataplane Reset. APB interface is not influenced by this signal, that is, apb can continue to read and write registers.
			6	-	-	Reserved
			5	-	-	Reserved
			4	-	-	Reserved
			[3:1]	-	-	Reserved
		Page Ready	0	0	W1SC	Write a '1' to this bit to inform the transmitter that a new page is ready for transmission in the page registers. The bit will clear when the transmit dataplane has transferred the page to its internal register, and the apb can then start writing a new page.

Table 5 • **Auto-Negotiation Tx Register**

Offset	Register Name	Bit(s) Name	Bit	Default	Action	Description
1	ACK Control	ACK.on	4	0	RWSC	When asserted the transmitter will insert ACK into the outgoing frame for the next ACK.cnt frames, after which time it will send the value in the Page register. The bit clears when the cnt is done. This bit is sampled by the dataplane when a new page is transmitted, that is, when Page.Ready goes from non-asserted to asserted.
		ACK.cnt	[3:0]	X	RW	The number of frames to send with ACK asserted. Count is minus 1, for example, writing 1 will send 2 frames with ACK asserted.
2	External Cfg	Link Control	7	0	RW	When asserted all links are disconnected from the MDI.
		C49	2	0	RW	Configures the transmitter to run as 10ge.
		Pause Port	0	0	RW	Configures the transmitter to run with port pause.
3			[3:0]	-	-	Reserved
8	Page[7:0]		-	x	RW	Page byte 0
9	Page[15:8]		-	x	RW	Page byte 1
10	Page[23:16]		-	x	RW	Page byte 2
11	Page[31:24]		-	x	RW	Page byte 3
12	Page[39:32]		-	x	RW	Page byte 4
13	Page[47:40]		-	x	RW	Page byte 5

The following table lists the Rx register functionalities of the APB.

Table 6 • **Auto-Negotiation Rx Register**

Offset	Register Name	Bit(s) Name	Bit	Default	Action	Description
0	Main Control	Reset	7	0	RW	Dataplane Reset. APB interface is not influenced by this signal, that is, apb can continue to read and write registers.
			6	0	RW	Reserved
		Interrupt Enable	5	0	RW	When asserted high the core will drive the interrupt line when Page. Ready is asserted.
			4	-	-	Reserved
		Lock	3	-	R	Dataplane Lock. The dataplane is locked to a valid Auto-Negotiation (Clause 73) frame. Same as an_receive_idle, except inverted.
		Bit Lock	2		R	Dataplane Bit Lock. Used for debug. It indicates that the incoming frame abides by the manchester rules.

Table 6 • **Auto-Negotiation Rx Register**

Offset	Register Name	Bit(s) Name	Bit	Default	Action	Description
		First	1		W	Write a '1' to this bit to force the Auto-Negotiation receiver to restart its matching engine. For example, if it has already delivered the incoming page, and the page continues to be received, writing '1' to this bit will allow another page to be received. Must set to 0 when not using this functionality.
		Page Ready	0	0	W1C	When asserted high, a page is ready in the page registers. While this signal is high, the dataplane will not make any changes to the Page Registers. Write 1 to clear.
1	ACK Control	ACK on	4	0	RW	The receiver hunts for three consecutive matching frames, when this bit is low the ACK is excluded from the match, when it is high it is included in the match and ACK is compare to 1'b1.
2	External Cfg/Status	Link Status	7	0	RW	When asserted the configured link is Good, that is, PCS Status is asserted.
		C49	2	0	RW	Configures the receiver to run as 10ge.
		Pause Port	0	0	RW	Configures the receiver to run with port pause.
3	Lane Cfg	Lane Number	[3:0]	0	RW	Identifies which lanes is used to receive Clause 73 frames.
8	Page[7:0]		-	x	R	Page[0] is the first bit received.
9	Page[15:8]		-	x	R	-
10	Page[23:16]		-	x	R	-
11	Page[31:24]		-	x	R	-
12	Page[39:32]		-	x	R	-
13	Page[47:40]		-	x	R	Page[47] is the last bit received

Table 7 • **Link Training Tx Register**

Offset	Register Name	Bit(s) Name	Bit	Default	Action	Description
0	Main Control	Reset	7	0	RW	Dataplane Reset. APB interface is not influenced by this signal, that is, apb can continue to read and write registers.
		Enable	6	0	RW	When asserted high (and external bypass is low), the core dataplane drives the MDI.
			5	-	-	Reserved
		External Bypass	4	-	R	An external element prohibits the core from owning the MDI.
			[3:1]	-	-	Reserved
		Page Ready	0	0	W1SC	Write a '1' to this bit to inform the transmitter that a new page is ready for transmission in the page registers. The bit will clear when the transmit dataplane has transferred the page to its internal registers, and the apb can then start writing a new page.
4	Page[0+:8]		-	x	RW	Corresponds to Status Report Field [7:0].
5	Page[8+:8]		-	x	RW	Corresponds to Status Report Field [15:8].
6	Page[16+:8]		-	x	RW	Corresponds to Coefficient Update Field [7:0].
7	Page[24+:8]		-	x	RW	Corresponds to Coefficient Update Field [15:8].

Table 8 • **Link Training Rx Register**

Offset	Register Name	Bit(s) Name	Bit	Default	Action	Description
0	Main Control	Reset	7	0	RW	Dataplane Reset. APB interface is not influenced by this signal, that is, apb can continue to read and write registers.
		Enable	6	0	RW	Has no function, reserved for future possibilities. If APB wants to turn off the receiver then just hold it in reset.
		Interrupt Enable	5		RW	When asserted high the core will drive the interrupt line when Page.Ready is asserted.
			4	-	-	Reserved
		Lock	3		R	Dataplane Lock. The dataplane is locked to a valid Clause 72 frame.
			2	-	-	Reserved
		First	1		W	Write a 1'b1 to force the receive to restart its matching engine. For example, if already delivered the incoming page, and the page continues to be received, writing this bit will another page to be received. Always reads as zero.
		Page Ready	0	0	W1C	When asserted high, a page is ready in the page registers. While this signal is high, the dataplane will not make any changes to the Page Registers. Write 1 to clear.
4	Page[0+:8]		-	-	R	Corresponds to Status Report Field [7:0].
5	Page[8+:8]		-	-	R	Corresponds to Status Report Field [15:8].
6	Page[16+:8]		-	-	R	Corresponds to Coefficient Update Field [7:0].
7	Page[24+:8]		-	-	R	Corresponds to Coefficient Update Field [15:8].

Table 9 • **Tx Ctrl Register**

Offset	Register Name	Bit(s) Name	Bit	Default	Action	Description
0	Main Control	PMA Data	[1:0]	0	RW	PMA Tx Data Select. Used to select the PMA Tx raw data from the Tx clause blocks to transmit. This field is controlled by the Auto Negotiation firmware. 0 = PCS sublayer for 10GE (C49) 1 = PCS sublayer for 1GE (C36) - currently not supported 2 = Auto-Negotiation for back plane (C73) 3 = PMD Sublayer for Link Training (C72)

Table 9 • **Tx Ctrl Register**

Offset	Register Name	Bit(s) Name	Bit	Default	Action	Description
1	Parameter Read	x	3	0	R	MAC Transmitter Loopback Local Enable, reports the GUI value of CFG_MAC_TX_LPBK_LOCAL_EN.
		x	2	0	R	Reserved
		x	1	0	-	Reserved
		x	0	0	-	Reports the GUI value of 10GBASE-R or 10GBASE-KR. 0 – 10GBASE-R 1 – 10GBASE-KR

Table 10 • **Rx Ctrl Register**

Offset	Register Name	Bit(s) Name	Bit	Default	Action	Description
0	Status Read	x	1	0	-	Reserved
		pcs49 status	0	0	R	The receive status signal for 10GE. This signal indicates that the receiver is in block lock and not in hi_ber state.
1	Parameter Read	x	3	0	R	MAC Receiver Loopback Local Enable, reports the GUI value of CFG_MAC_RX_LPBK_LOCAL_EN
		x	2	0	-	Reserved
		x	1	0	-	Reserved
		x	0	0	R	Reports the GUI value of 10GBASE-R or 10GBASE-KR 0 – 10GBASE-R 1 – 10GBASE-KR

Table 11 • MAC Tx Config Register

Offset	Register Name	Bit(s) Name	Bit	Default	Action	Description
0	MAC Static	static_mac_tx_ifg_cnt	[13:8]	12	RW	Configure Tx IFG Count. This signal configures the IFG amount. The standard is 12, but the core supports other values. The minimum supported IFG value is 1, and the maximum is 48. Values less than 1 will behave as 1, and values above 48 will behave as 48. The signal is static and is used when GUI parameter CFG_MAC_TX_IFG_CNT is set to 0.
		static_mac_tx_ifg_dic_mode_en	4	1	RW	Configure Tx DIC Mode. This signal enables the IFG spacing to be performed with regards to DIC as definite by the IEEE specifications. When the signal is asserted high, DIC is enabled, when it is asserted low DIC is disabled.The signal is static and is used when GUI parameter CFG_MAC_TX_IFG_CNT is set to 0.
		static_mac_tx_preamble_en	3	0	RW	Configures the Tx core to use a customer preamble. The signal is valid when GUI parameter CFG_MAC_TX_PREAMBLE is set to 1.
		static_mac_tx_preamble_wcm1	[2:0]	SWC	RW	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 GUI parameter CFG_MAC_TX_PREAMBLE is set to 1 and its initial value is set to parameter SWC.
1	Pause MAC Address 1	pause_tx_mac_addr[31:0]	[31:0]	0	RW	The source MAC address inserted in pause frames. This signal is active when GUI parameter CFG_PAUSE_TX_NEW=1.

Table 11 • **MAC Tx Config Register**

Offset	Register Name	Bit(s) Name	Bit	Default	Action	Description
2	Pause MAC Address 2	pause_tx_mac_addr[47:32]	[15:0]	0	RW	The source MAC address inserted in pause frames. This signal is active when GUI parameter CFG_PAUSE_TX_NEW is set to 1.
3	Config	cfg_sys_mac_tx_fifo_paf	[20:17]	0	RW	Configure Tx FIFO Programmable Almost Full level. The signal configures the threshold. This signal can be changed dynamically.
		cfg_sys_mac_tx_en	16	1	RW	Configure Tx Enable. This configuration signal enables the MAC TX core to send frames onto the line. When this signal is asserted low, no data-frames will be sent. When the signal is asserted high, data flows normally. The signal is sampled on a packet boundary, that is, no partial packets will be generated as a consequence of changing the assertion of this signal. The signal can be changed dynamically.
		mac_tx_max_pkt_len	[15:0]	0xC000	RW	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. This signal can be changed dynamically.

Table 11 • **MAC Tx Config Register**

Offset	Register Name	Bit(s) Name	Bit	Default	Action	Description
4	System 1	sys_mac_tx_fcs_ins	8	0	RW	Tx FCS Insert. This signal indicates if the core should insert FCS on all packets
		sys_mac_tx_fcs_err	7	0	RW	Tx FCS Error. This signal indicates if the core should insert an FCS error on the packet.. The FCS error is inserted by xoring the correct FCS with 0x5555_5555.
		sys_mac_tx_fcs_stomp	6	0	RW	Tx FCS Stomp. This signal indicates if the core should insert an FCS stomp The FCS stomp is inserted by xoring the correct FCS with 0xFFFF_FFFF.
		sys_mac_tx_ifg_cnt	[5:0]	0	RW	Tx Per Packet IFG. This signal is only used if GUI parameter CFG_MAC_TX_IFG_CNT is greater than 0.
5	System 2	mac_tx_preamble[31:0]	[31:0]	0	RW	Tx Preamble lower 32bit. This field is only used for a custom preamble when GUI parameter CFG_MAC_TX_PREAMBLE and mac_tx_preamble_en bit of MAC Tx Config Register is set to 1. Required when SWC = 1 or 2
6	System 3	mac_tx_preamble[63:32]	[31:0]	0	RW	Tx Preamble upper 32bits. This field is only used for a custom preamble when GUI parameter CFG_MAC_TX_PREAMBLE and mac_tx_preamble_en bit of MAC Tx Config Register is set to 1. **Note:** Only required when SWC = 2

Table 12 • **MAC Rx Config Register**

Offset	Register Name	Bit(s) Name	Bit	Default	Action	Description
0	MAC Static	mac_rx_fcs_remove	3	0	RW	Configure Rx FCS Remove. This bit configures the core to remove the FCS field. When asserted high the core strips the FCS.
		mac_rx_preamble_wcm1	[2:0]	1	RW	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 GUI parameter CFG_MAC_RX_PREAMBLE is set to 1.
1	Pause MAC Address 1	pause_rx_mac_addr[31:0]	[31:0]	0	RW	The destination MAC address used for identification of pause uni-cast frames. This field is active when parameter GUI CFG_PAUSE_RX_PORT or CFG_PAUSE_RX_PFC is set to 1.
2	Pause MAC Address 2	pause_rx_mac_addr[47:32]	[15:0]	0	RW	The destination MAC address used for identification of pause uni-cast frames. This field is active when GUI parameter CFG_PAUSE_RX_PORT or CFG_PAUSE_RX_PFC is set to 1.

Table 12 • **MAC Rx Config Register**

Offset	Register Name	Bit(s) Name	Bit	Default	Action	Description
3	Config	sys_mac_rx_lpbk_local_en	17	0	RW	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.
		cfg_sys_mac_rx_en	16	1	RW	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. The bit can be changed dynamically.
		sys_mac_rx_max_pkt_len	[15:0]	0xC000	RW	Configure Rx Maximum Packet Length. A valid packet is less than or equal to this value. If a larger packet is received it will be delivered 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. This bit can be changed dynamically.
4	System 1	sys_mac_rx_gfc	0	0	RW	Rx Global Flow Control. When this bit is '1' the core will not read from the receive FIFO, and when the bit is '0' 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. The signal is normally tied low

Table 13 • **MAC Tx Static Register**

Offset	Register Name	Bit(s) Name	Bit	Default	Action	Description
12	pkt_ok	stats_pkt_ok_cnt	[31:0]	0	R	Packet error free counter
11	pkt_pad	stats_pkt_pad_cnt	[31:0]	0	R	Padded packet counter

Table 13 • **MAC Tx Static Register**

Offset	Register Name	Bit(s) Name	Bit	Default	Action	Description
10	pkt_vlan	stats_pkt_vlan_cnt	[31:0]	0	R	VLAN packet counter
9	pkt_control	stats_pkt_control_cnt	[31:0]	0	R	Control packet counter
8	pause	stats_pkt_pause_cnt	[31:0]	0	R	Pause packet counter
7	multicast	stats_pkt_multicast_cnt	[31:0]	0	R	Multicast packet counter
6	broadcast	stats_pkt_broadcast_cnt	[31:0]	0	R	Broadcast packet counter
5	err	stats_pkt_err_cnt_cnt	[31:0]	0	R	Errored packet counter
4	err_frm	stats_pkt_err_frm_cnt	[31:0]	0	R	Errored at framing counter
3	err_fcs	stats_pkt_err_fcs_cnt	[31:0]	0	R	Errored at FCS counter
2	err_len_short	stats_pkt_err_len_short_cnt	[31:0]	0	R	Errored at length, with short
1	err_len_check	stats_pkt_err_len_check_cnt	[31:0]	0	R	Errored at length, with check counter
0	err_len_long	stats_pkt_err_len_long_cnt	[31:0]	0	R	Errored at length, with long counter

Table 14 • **MAC Rx Static Register**

Offset	Register Name	Bit(s) Name	Bit	Default	Action	Description
12	pkt_ok	stats_pkt_ok_cnt	[31:0]	0	R	Packet error free counter
11	pkt_pad	stats_pkt_pad_cnt	[31:0]	0	R	Padded packet counter
10	pkt_vlan	stats_pkt_vlan_cnt	[31:0]	0	R	VLAN packet counter
9	pkt_control	stats_pkt_control_cnt	[31:0]	0	R	Control packet counter
8	pause	stats_pkt_pause_cnt	[31:0]	0	R	Pause packet counter
7	multicast	stats_pkt_multicast_cnt	[31:0]	0	R	Multicast packet counter
6	broadcast	stats_pkt_broadcast_cnt	[31:0]	0	R	Broadcast packet counter
5	err	stats_pkt_err_cnt_cnt	[31:0]	0	R	Errored packet counter
4	err_frm	stats_pkt_err_frm_cnt	[31:0]	0	R	Errored at framing counter
3	err_fcs	stats_pkt_err_fcs_cnt	[31:0]	0	R	Errored at FCS counter
2	err_len_short	stats_pkt_err_len_short_cnt	[31:0]	0	R	Errored at length, with short
1	err_len_check	stats_pkt_err_len_check_cnt	[31:0]	0	R	Errored at length, with check counter
0	err_len_long	stats_pkt_err_len_long_cnt	[31:0]	0	R	Errored at length, with long counter

4.3 Nomenclature

This section provides detail on a number of specific nomenclatures.

4.3.1 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 is used to reset the Tx blocks and I_SYS_RX_SRESET is used to reset the Rx blocks.

4.3.2 Dynamic Configuration Signals

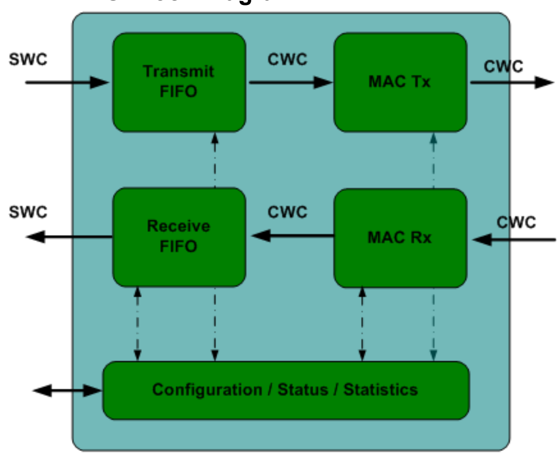
The core has a number of dynamic configuration signals. These signals can be changed at any time.

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

4.4 Ethernet MAC/RS Overview

The high level architecture of the Ethernet MAC/RS is as shown in the following figure.

Figure 4 • **Ethernet TX/Rx FIFO Block Diagram**



4.4.1 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 will either be dropped, or passed with an error. The error will be transmitted by the MAC as an FCS error.

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

4.4.2 Receive FIFO

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

4.4.3 MAC Tx

The Transmit MAC performs the following:

- 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

4.4.4 MAC Rx

The Receive MAC performs the following:

- Receives data from 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

4.4.5 Statistics

The transmit and receive statistics are made 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 in the configuration GUI sections MAC Tx Counters and MAC Rx Counters.

4.5 Ethernet Interface

This section expands on the different interfaces, presents timing diagrams and documents the bus conventions.

4.5.1 Tx Dataplane

4.5.1.1 Tx Dataplane Signal Encoding

The Tx Dataplane Bus Protocol Encoding is listed in the following table. An 8 byte system bus is used as an example.

Table 15 • **Tx Dataplane Bus Protocol Encoding**

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 associated encoding of I_SYS_MAC_TX_BC is listed in the following table.

Table 16 • **Encoding of I_SYS_MAC_TX_BC**

I_SYS_TX_BC[2:0]	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

4.5.1.2 Tx Dataplane Interface Errors

The tx core performs complete error checks on the delivered data. Three types of errors can be introduced by the user, that is:

- **Protocol error:** Errors in the transmit FIFO interface usage.
- **FIFO Overflow error:** The FIFO becomes full during the transmission of a package. This will never happen if the user abides by the FIFO flags, but the core handles the event gracefully.
- **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 will cause the error or loss of one or more packets. The user should drive the interface so as not to introduce any of these errors.

4.5.1.3 Tx Dataplane Padding

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

- The packet was delivered as a short packet.
- The packet became short because of a FIFO over flow event, using the O_SYS_MAC_TX_FIFO_AF output signal to determine how full the FIFO is can help avoid this situation from ever happening.

4.5.2 Rx Dataplane

4.5.2.1 Rx Dataplane Signal Encoding

The Rx Dataplane Bus Protocol Encoding is listed in the following table. An 8 byte system bus is used as an example.

Table 17 • **Rx Dataplane Bus Protocol Encoding**

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 associated encoding of O_SYS_MAC_RX_BC is listed in the following table.

Table 18 • **Encoding of O_SYS_MAC_RX_BC**

O_SYS_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

4.5.2.2 Rx Dataplane Interface Errors

The rx core performs complete error checks on the delivered data. Three types of errors can be introduced by the user, that is:

- **FIFO Overflow error:** The FIFO becomes full during the transmission of a package. This will never happen if the user abides by the FIFO flags, but the core handles the event gracefully.
- **Rx Packet Errors:** These errors will 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 will cause the error or loss of one or more packets. The user should 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 will handle the event gracefully. The core does the following on FIFO overflow:

- If the overflow happens in the middle of the packet the packet is colored bad internally, overflow flag asserts, 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 complete packet is dropped.

The dropped data is not accounted for in any of the statistics. If the user wants all data accounted for, then the FIFO should be driven in such a way as to not loose data. This can be done by synchronising clocks and increasing the FIFO depth if required.

4.5.3 System Interface Remote Loopback

A system interface remote loopback (for testing purposes) can be implemented by looping the receive system bus directly to the transmit system bus. In this mode the tx playout margin set to 4 to allow for the crossing of the receive asynchronous FIFO.

Connect the bus signals as follows:

- 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'

The connections depict the scenario where the receiver is configured to remove the CRC. If the receiver is not configured to remove the CRC, then set MAC Tx Config Register bit sys_mac_tx_fcs_ins to '0'.

4.5.4 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 parameters CFG_MAC_TX_LPBK_LOCAL_EN and CFG_MAC_RX_LPBK_LOCAL_EN must be enabled to expose the Tx\Rx local loopback interface. In this mode the tx playout margin set to 4 to allow for the crossing of the receive asynchronous FIFO.

Connect the bus signals as follows:

- 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: The signal can be changed dynamically, but errors should be expected during the transition, especially if data is flowing when the signal is changed.

4.5.5 Pause Interface

The core supports the transmission of pause frames, and the reception and procession of pause frames. Pause-port is completely contained between the transmit and receive cores, that is, the reception of a pause-port frame with a non-zero pause time, will cause the transmit MAC to stop transmission, if enable to do so. The output signals O_PAUSE_TX_PORT_ON, O_SYS_PAUSE_RX_PORT_XOFF, and O_SYS_PAUSE_RX_PFC_XOFF_W are used to determine when this behavior is occurring. For pause-pfc, the core receives and processes the frames, but the xoff signal needs to be acted upon external to the transmit core, since the transmit MAC only has one system FIFO queue.

4.5.5.1 Pause Tx

The source MAC address for pause frames is sourced from static_pause_tx_mac_addr MAC Tx Config Register.

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:

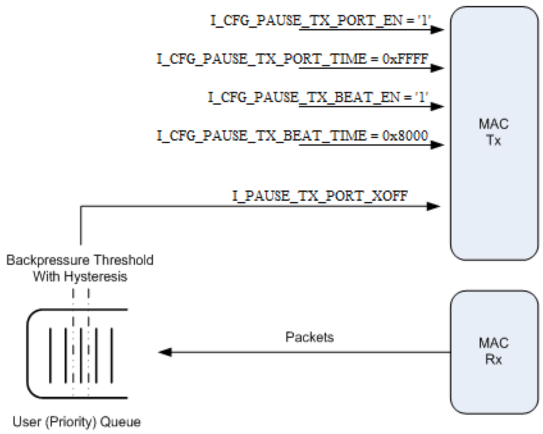
- The user drives I_CFG_PAUSE_TX_SEND_STRB high.
- A transition happens on I_PAUSE_TX_PORT_XOFF or I_PAUSE_TX_PFC_XOFF_W.
- I_CFG_PAUSE_TX_BEAT_EN is enabled and I_CFG_PAUSE_TX_BEAT_TIME has expired.

The pause port is active when parameter CFG_PAUSE_TX_NEW is ‘1’.

4.5.5.1.1 Example of Tx Configuration

The following figure shows the most common Tx configuration.

Figure 5 • **Pause Tx Example Configuration**



When the user queue crosses its defined backpressure threshold, the user should 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 should 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 I_CFG_PAUSE_TX_BEAT_TIME quantas, then another pause frame is sent with pause quanta of 16’hffff.

The same usage applies to all PFC flows.

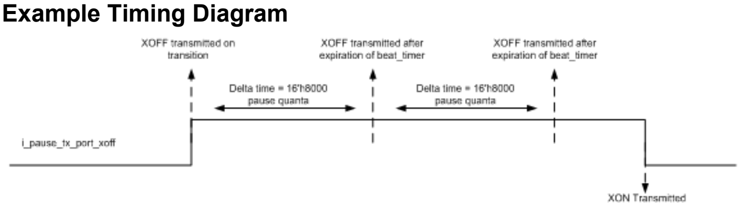
Normally the configuration signals are configured as follows:

- 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

4.5.5.1.2 Example of Tx Timing Diagram

The action diagram for the example configuration is depicted.

Figure 6 • **Pause Tx Example Timing Diagram**



A pause frame is transmitted every 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 every time I_CFG_PAUSE_TX_BEAT_TIME expires.

The beat time is internal to the core, the user supplies the expiration value on I_CFG_PAUSE_TX_BEAT_TIME.

4.5.5.2 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.

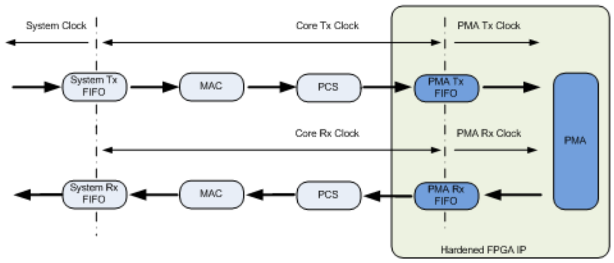
The core is enabled to process pause frames if I_CFG_SYS_PAUSE_RX_PORT_EN or I_CFG_SYS_PAUSE_RX_PFC_EN_W is asserted.

When a valid pause frame is received, the timer is latched, decremented, and while non-zero, the associated xoff signals is asserted.

4.5.6 Clocking & Resets

The PMA FIFO is typically hardened inside the FPGA SerDes, and implemented as a phase alignment FIFO. The resulting clocking architecture is as follows.

Figure 7 • **FPGA Clocking Architecture**



The PMA Tx Clock is +/- xxxx PPM of the PMA Rx Clock.

The Core Tx Clock is the same frequency as the PMA Tx Clock, but different phase.

The Core Rx Clock is the same frequency as the PMA Rx Clock, but different phase.

The System Tx FIFO is configured as asynchronous.

The System Rx FIFO is always asynchronous in this configuration.

4.5.6.1 Clock Desciption

I_SYS_CLK: The system clock decouples the user clock domain from the core clock domains. The clock can be driven with the same clock as is driving I_CORE_TX_CLK or I_CORE_RX_CLK, or it can be driven with a completely different clock. For wire speed throughput this clock should be the same as I_CORE_TX_CLK, this will provide the lowest latency.

I_CORE_TX_CLK: The Tx clock is used to transmit data. This is driven by the transceivers tx clock. The frequency is implementation specific.For 10GE requires 32bit @ 322.265625MHz or 64bit @ 161.1328125MHz.

I_CORE_RX_CLK: The RX clock is used to receive data. This is driven by the transceivers rx clock. The frequency is implementation specific. For 10GE requires 32bit @ 322.265625MHz or 64bit @ 161.1328125MHz.

4.5.7 Initialization

The core is initialized through its APB interface (see APB Control Registers section for more details). 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 assertion.

The tx fifo flag, that is, O_SYS_MAC_TX_FIFO_AF will assert 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 complete the flags will de-assert.

4.5.8 Statistics Vector

The MAC statistics are delivered through a statistics vector or optionally as APB registers. This vector can be used to generate the desired RMON statistics. This can be done by the user.

The encoding of the vector is listed in the following table. The receive and transmit vectors are symmetrical.

Table 19 • **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 is used to qualify 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	The length of the packet.

4.5.8.1 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 was flagged in error by the MAC delineation process. This happens if the PCS flagged the packet in error with /E/, or if the packet was corrupted so as not to allow for proper delineation.

4.5.8.2 Error on Length Short

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

4.5.8.3 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.

This function can be disabled by de-asserting CFG_MAC_[RX,TX]_CHECK_LT.

4.5.8.4 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.

4.5.9 Reconciliation Sublayer

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

Table 20 • **Priority of the Fault State Generation**

Priority	Fault
0	Local
1	Remote
2	Idle

5 Interface Description

5.1 Configuration GUI Parameters

Core10GMAC has GUI parameters for configuring the core, as listed in the following table.

Table 21 • **GUI Parameters**

Name	Range	Default Value	Description
PERSONALITY			
TYPE_10G	0 to 2	1	Selects 10G Type: 0: 10GBASE-R 1: 10GBASE-KR 2: XGMII
FAMILY	26	26	Must be set to the required FPGA family: 26: PolarFire
SWC	1, 2	2	System Word Count. A word is 4 bytes, and valid values are 1 or 2. 1 creates a system bus with 4 bytes, and 2 creates a system bus with 8 bytes, that is, 32bits or 64 bits
CWC	1, 2	1	Core Word Count. The buswidth of the internal core measured in words. A word is 4 bytes, and valid values are 1 or 2. 1 creates a system bus with 4 bytes, and 2 creates a system bus with 8 bytes, that is, 32bits or 64 bits
MAC FEATURE			
CFG_MAC_TX_FIFO_DEPTH		32	The Depth of the system FIFO. Normally set to 32 entries.
CFG_MAC_TX_PREAMBLE	0 to 1	0	When '1' the core is enabled to send custom preamble and the associated signals are active. When ‘0’ the function is disabled.

Table 21 • GUI Parameters

Name	Range	Default Value	Description
CFG_MAC_TX_IFG_CNT	0 to 48	12	Configure Tx IFG Count. This signal configures the IFG amount. The standard is 12, but the core supports other values. The minimum supported IFG value is 1, and the maximum is 48. When '0' Tx IFG Count is dynamic, the value on sys_mac_tx_ifg_cnt MAC Tx Config Register is sampled on I_SYS_MAC_TX_EOP and used for the associated packet. Default value stored is 12. When 1 to 48 Tx IFG Count is fixed to this parameters value. For example: 1: Fixed at 1 2: Fixed at 2 … 47: Fixed at 47 48: Fixed at 48
CFG_MAC_TX_CHECK_LT	0 to 1	0	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 no consequence for the data-path, that is, it only influences the statistics reporting.
CFG_MAC_TX_LPBK_LOCAL_EN	0 to 1	0	When ‘1’ the core supports local loopback. When ‘0’ the core does not support local loopback.
CFG_MAC_RX_FIFO_DEPTH		32	The Depth of the system FIFO. Normally set to 32 entries.
CFG_MAC_RX_PREAMBLE	0 to 1	0	When ‘1’ the core is enabled to receive custom preamble and the associated signals are active. When ‘0’ the function is disabled.
CFG_MAC_RX_CHECK_LT	0 to 1	0	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 to 1	0	When ‘1’ the core supports local loopback. When ‘0’ the core does not support local loopback.

Table 21 • GUI Parameters

Name	Range	Default Value	Description
CFG_SYS_MAC_RX_GFC_SELECT	0 to 1	0	Rx Global Flow Control Select When '1' I_SYS_MAC_RX_GFC input is used. When '0' the MAC Rx Config Register sys_mac_rx_gfc is used. When Rx Global Flow Control is asserted high the core will 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 to 1	0	MAC TX Static Packet Counters Enable - These counters are used to monitor the Static Packet signals from the Static Vector output by incrementing their values on every occurrence.
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 to 1	0	MAC RX Static Packet Counters Enable - These counters are used to monitor the Static Packet signals from the Static Vector output by incrementing their values on every occurrence.
PAUSE FEATURES			
CFG_PAUSE_TX_EN	0 to 3	0	When '0' the core support for pause is disabled. When '1' the core is enabled to support pause port. When '2' the core is enabled to support pause pfc. When '3' the core is enabled to support pause port and pause pfc.
CFG_PAUSE_TX_TIMER	0 to 1	0	When '1' the core is enabled to support pause timer on transmit, that the ability to repeat send pause frames on a user defined beat. When '0' the function is not supported.

Table 21 • GUI Parameters

Name	Range	Default Value	Description
CFG_PAUSE_RX_EN	0 to 3	0	When '0' the core support for pause is disabled. When '1' the core is enabled to support pause port. When '2' the core is enabled to support pause pfc. When '3' the core is enabled to support pause port and pause pfc.
CFG_PAUSE_RX_CHECK_PMC	0 to 1	1	When ‘1’ the core is enabled to use pause multi-cast address to identify pause frames. When ‘0’ the function is not supported.
CFG_PAUSE_RX_CHECK_PUC	0 to 1	0	When ‘1’ the core is enabled to use pause unicast-cast address to identify pause frames. When ‘0’ the function is not supported.
PCS 73 Rx Gearbox			
CFG_PCS73_RX_GRBX_EN	0 to 1	1	PCS Auto Negotiation Receiver Gearbox Enable. 0 : Disable, (that is, bypassed) 1 : Enable
APB Timeout			
APB_TIMEOUT_EN	0 to 1	0	APB Timeout Enable. When ‘1’ the core is enabled to use timeout value configured for "APB_TIMEOUT_COUNT" parameter to release PREADY of APB interface. When ‘0’ the core ignores the "APB_TIMEOUT_COUNT" parameter value.
APB_TIMEOUT_COUNT	0 to 255	80	APB Timeout Count. This parameter configures timeout value When APB_TIMEOUT_EN is ‘1’. If PREADY of APB interface remains low for the timeout value then it will be released. To count the timeout value, PCLK of APB interface is used.

5.2 I/O Signals

The port signals for the Core10GMAC macro are listed in the following table.

Table 22 • **Port Signals for Core10GMAC Macro**

Name	Width	Direction	Description
SYSTEM			
I_SYS_CLK	1	In	The clock for the system side clock domain.
I_SYS_TX_SRESET	1	In	Synchronous reset of the transmit core, with reference to the I_SYS_CLK domain.
I_SYS_RX_SRESET	1	In	Synchronous reset of the receive core, with reference to the I_SYS_CLK domain.
CORE			
I_CORE_TX_CLK	1	In	The transmit clock for the line side clock domain.
O_CORE_TX_SRESET	1	Out	The internally generated reset signal. Can be used to reset other modules.
I_CORE_RX_CLK	1	In	The receive clock for the line side clock domain.
O_CORE_RX_SRESET	1	Out	The internally generated reset signal. Can be used to reset other modules.
MAC TX PACKET INTERFACE			
O_SYS_MAC_TX_RDY	1	Out	Tx Ready. This signal indicates that the MAC is done with its reset initialization and ready to transmit data. Writes to the transmit FIFO will be dropped when this signal is asserted. **Note:** When this signal is asserted, the transmit FIFO flags will all be asserted.
O_SYS_MAC_TX_FAULT	1	Out	Tx Fault. This signal indicates that the transmit MAC is in an RS fault state, and that packets which are written to the transmit FIFO will be dropped. This signal does not have any interaction with the FIFO flags.
O_SYS_MAC_TX_FIFO_AF	1	Out	Tx FIFO Almost Full Flag. This signal indicates that the transmit FIFO is almost full. When the signal transitions from de-asserted to asserted the FIFO has space for four more entries.
I_SYS_MAC_TX_EN	1	In	Tx Enable. This signal is used to qualify the writes to the Tx FIFO. The value of the other Tx data path input signals is immaterial when this signal is not asserted.
I_SYS_MAC_TX_SOP	1	In	Tx Start of Packet. This signal indicates the start of packet event.

Table 22 • Port Signals for Core10GMAC Macro

Name	Width	Direction	Description
I_SYS_MAC_TX_EOP	1	In	Tx End of Packet. This signal indicates the end of packet event.
I_SYS_MAC_TX_BC	8	In	Tx End of Packet Byte Count. This signal indicates the number of valid byte entries in the transfer. The formatting is described in the interface section. The signal is sampled when: I_SYS_MAC_TX_EN and I_SYS_MAC_TX_EOP are asserted.
I_SYS_MAC_TX_DATA	32 or 64	In	Tx Data. This signal contains the data for transfer. The signal is sampled when I_SYS_MAC_TX_EN is asserted. The formatting is described in the interface section.
O_SYS_MAC_TX_ERR_BUS_PROTOCOL	1	Out	Transmit Bus Protocol Error. This signal asserts for one cycle when an error has been discovered with the user protocol on the Transmit data plane interface, for example, if two sop are received without an eop in between.
O_SYS_MAC_TX_ERR_FIFO_OVERFLOW	1	Out	Transmit FIFO Overflow Error. This signal asserts for one cycle on a Tx FIFO overflow event. This will never happen if the user abides by the FIFO flags, but the core handles the event gracefully.
O_SYS_MAC_TX_ERR_FIFO_UNDERRUN	1	Out	Transmit FIFO Underrun Error. This signal asserts for one cycle on a Tx FIFO underrun event. The core will play 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	40	Out	The Transmit Statistics bus.
MAC RX PACKET INTERFACE			
I_SYS_MAC_RX_GFC	1	In	Rx Global Flow Control. When this signal is asserted high the core will not read from the receive FIFO, and when the signal is asserted 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. The signal is normally tied low. This signal is only available when CFG_SYS_MAC_RX_GFC_SELECT is enabled, otherwise the MAC_RX_CONFIG register sys_mac_rx_gfc is used.

Microsemi a Microchip company

Table 22 • **Port Signals for Core10GMAC Macro**

Name	Width	Direction	Description
O_SYS_MAC_RX_RDY	1	Out	Rx Core Ready. This signal is asserted low while the receive core is going through reset and initialization. When asserted high the core is ready to play.
O_SYS_MAC_RX_EN	1	Out	Rx Enable. This signal is used to qualify the received data. The values of the other Rx data path output signals are undefined when this signal is not asserted.
O_SYS_MAC_RX_SOP	1	Out	Rx Start of Packet. This signal indicates the start of packet event. Note, this signal will always be asserted low when O_SYS_MAC_RX_EN is deasserted.
O_SYS_MAC_RX_EOP	1	Out	Rx End of Packet. Note, this signal will always be asserted low when O_SYS_MAC_RX_EN is deasserted.
O_SYS_MAC_RX_ERR	1	Out	Rx Error. This signal indicates an error with the associated packet. The signal will 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. **Note:** This signal will always be asserted low when O_SYS_MAC_RX_EN is deasserted.
O_SYS_MAC_RX_ERR_W[7]	1	Out	Not used
O_SYS_MAC_RX_ERR_W[6]	1	Out	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 is asserted. When this signal asserts O_SYS_MAC_RX_ERR will also assert.
O_SYS_MAC_RX_ERR_W[5]	1	Out	Rx CRC Error. This signal indicates an FCS error with the associated packet. The signal will only assert when O_SYS_MAC_RX_EOP is asserted. When this signal asserts O_SYS_MAC_RX_ERR will also assert. The signal is active when CFG_MAC_RX_CRC is asserted.

Table 22 • **Port Signals for Core10GMAC Macro**

Name	Width	Direction	Description
O_SYS_MAC_RX_ERR_W[4]	1	Out	Rx FCS Stomp Error. This signal indicates an FCS stomp with the associated packet. The signal will only assert when O_SYS_MAC_RX_EOP is asserted. When this signal asserts O_SYS_MAC_RX_ERR will also assert. **Note:** A stomped packet will also be flagged as a CRC errored packet. Hence, this signal can be ignored by most users. The signal is active when CFG_MAC_RX_CRC is asserted.
O_SYS_MAC_RX_ERR_W[3]	1	Out	Rx Short Frame Error. This signal indicates that the packet is less than 64 bytes long. The signal will only assert when O_SYS_MAC_RX_EOP is asserted. When this signal asserts O_SYS_MAC_RX_ERR will also assert. This signal is active when CFG_MAC_RX_CHECK_LEN_SHORT is asserted.
O_SYS_MAC_RX_ERR_W[2]	1	Out	Rx Long Frame Error. This signal indicates that the packet is larger than I_CFG_SYS_MAC_RX_MAX_PKT_LEN. The signal asserts when the length exceeds the maximum configured, and stays asserted until O_SYS_MAC_RX_EOP is asserted. When this signal asserts O_SYS_MAC_RX_ERR will also assert. This signal is active when CFG_MAC_RX_CHECK_LEN_LONGis asserted.
O_SYS_MAC_RX_ERR_W[1]	1	Out	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 will also assert. This signal is active when CFG_MAC_RX_CHECK_LT is asserted.
O_SYS_MAC_RX_ERR_W[0]	1	Out	Rx Pause Frame Error. This signal indicates that the packet was flagged by the core as a pause frame. The signal will only assert when O_SYS_MAC_RX_EOP is asserted. When this signal asserts O_SYS_MAC_RX_ERR will also assert. This signal is active when CFG_PAUSE_RX_PORT is asserted.
O_SYS_MAC_RX_BC	8	Out	Rx Byte Count. This signal indicates the number of valid byte entries in the transfer. The formatting is described in the interface section.

Table 22 • **Port Signals for Core10GMAC Macro**

Name	Width	Direction	Description
O_SYS_MAC_RX_PREAMBLE	32 or 64	Out	Rx Preamble. This signal contains the receive preamble. The signal is valid when O_SYS_MAC_RX_EN and O_SYS_MAC_RX_SOP are asserted.
O_SYS_MAC_RX_DATA	32 or 64	Out	Rx Data. This signal contains the received data. The signal is only valid when O_SYS_MAC_RX_EN is asserted. The formatting is described in the interface section.
O_SYS_MAC_RX_ERR_FIFO_OVERFLOW	1	Out	Receive FIFO Overflow Error. This signal asserts for one cycle on an Rx FIFO overflow event. This only happens if the I_SYS_CLK is not running fast enough.The core will error a packet, which is presently being written to the FIFO, and it will completely drop a new packet while the FIFO is full. The end result being that the user will continue to see consistent packet boundaries on the system side.
O_SYS_MAC_RX_STATS_VECTOR	40	Out	The Receive Statistics bus. The encoding is documented in the statistics section.
MAC LOOPBACK INTERFACE			
O_MAC_TX_LPBK_LOCAL_CLK	1	Out	Local loopback ports. Connect to equivalent input signals on MAC Rx. This signal is active when CFG_MAC_TX_LPBK_LOCAL_EN is asserted.
O_MAC_TX_LPBK_LOCAL_CALL	32	Out	Local loopback ports. Connect to equivalent input signals on MAC Rx. This signal is active when CFG_MAC_TX_LPBK_LOCAL_EN is asserted.
O_MAC_TX_LPBK_LOCAL_CTRL_W	32 or 64	Out	Local loopback ports. Connect to equivalent input signals on MAC Rx. This signal is active when CFG_MAC_TX_LPBK_LOCAL_EN is asserted.
O_MAC_TX_LPBK_LOCAL_DATA_W	32 or 64	Out	Local loopback ports. Connect to equivalent input signals on MAC Rx. This signal is active when CFG_MAC_TX_LPBK_LOCAL_EN is asserted.
I_MAC_RX_LPBK_LOCAL_CLK	1	In	Local loopback ports. Connect to equivalent input signals on MAC Tx. This signal is active when CFG_MAC_RX_LPBK_LOCAL_EN is asserted.

Table 22 • **Port Signals for Core10GMAC Macro**

Name	Width	Direction	Description
I_MAC_RX_LPBK_LOCAL_CALL	32	In	Local loopback ports. Connect to equivalent input signals on MAC Tx. This signal is active when CFG_MAC_RX_LPBK_LOCAL_EN is asserted.
I_MAC_RX_LPBK_LOCAL_CTRL_W	32 or 64	In	Local loopback ports. Connect to equivalent input signals on MAC Tx. This signal is active when CFG_MAC_RX_LPBK_LOCAL_EN is asserted.
I_MAC_RX_LPBK_LOCAL_DATA_W	32 or 64	In	Local loopback ports. Connect to equivalent input signals on MAC Tx. This signal is active when CFG_MAC_RX_LPBK_LOCAL_EN is asserted.
TX PAUSE SIGNALS			
I_CFG_PAUSE_TX_SEND_STRB	1	In	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 CFG_PAUSE_TX_NEW is enabled.
I_CFG_PAUSE_TX_BEAT_EN	1	In	When asserted the core is enabled to continuously generate and send pause frames. The signal is enabled with CFG_PAUSE_TX_NEW and CFG_PAUSE_TX_TIMER are enabled.
I_CFG_PAUSE_TX_BEAT_TIME	16	In	The time interval that us used for the transmission beat. The timer is measured in 512 UI increments. The signal is enabled with CFG_PAUSE_TX_NEW and CFG_PAUSE_TX_TIMER are enabled.
I_CFG_PAUSE_TX_PORT_EN	1	In	When asserted the core is enabled to send pause-port frames. The signal is enabled with CFG_PAUSE_TX_NEW is enabled.
I_CFG_PAUSE_TX_PORT_TIME	16	In	The timer value inserted in pause-port frames. The signal is enabled with CFG_PAUSE_TX_NEW is enabled.
I_CFG_PAUSE_TX_PFC_EN_W	8	In	When asserted the core is enabled to send pause-pfc frames. *_W[0] is associated with priority 0. The signal is enabled with CFG_PAUSE_TX_NEW is enabled.
I_CFG_PAUSE_TX_PFC_TIME_W	128	In	The timer value inserted in pause-port frames. *_W[0+:16] is associated with priority 0. The signal is enabled with CFG_PAUSE_TX_NEW is enabled.

Table 22 • **Port Signals for Core10GMAC Macro**

Name	Width	Direction	Description
I_PAUSE_TX_PORT_XOFF	1	In	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 signals is enabled when CFG_PAUSE_TX_NEW is enabled, and when I_CFG_PAUSE_TX_PORT_EN is asserted.
I_PAUSE_TX_PFC_XOFF_W	8	In	Data-plane signal requesting pause-pfc. A transition on this signal causes the transmission of a pause-port frame. _W[0] is associated with priority 0. When _W[x] 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].The signals is enabled when CFG_PAUSE_TX_NEW is enabled, and when I_CFG_PAUSE_TX_PFC_EN_W[x] is asserted.
I_PAUSE_TX_PORT_REQ	1	In	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	1	Out	Indicates that the transmit MAC has halted packet transmission due to request on I_PAUSE_TX_PORT_REQ.
RX PAUSE SIGNALS			
I_SYS_PAUSE_RX_SRESET	1	In	Reset of the pause Rx block.
I_CFG_SYS_PAUSE_RX_PORT_EN	1	In	When asserted the core is enabled to process pause-port frames. This signal is active when CFG_PAUSE_RX_PORT is enabled.
I_CFG_SYS_PAUSE_RX_PFC_EN_W	8	In	When asserted the core is enabled to process pause-pfc frames. This signal is active when CFG_PAUSE_RX_PFC is enabled.
O_SYS_PAUSE_RX_PORT_XOFF	1	Out	The signal asserts when the core has received and processed a pause-port frame with an active pause time. This signal is active when CFG_PAUSE_RX_PORT is enabled
O_SYS_PAUSE_RX_PFC_XOFF_W	8	Out	The signal asserts when the core has received and processed a pause-pfc frame with an active pause time. *_W[0] is associated with priority 0.This signal is active when CFG_PAUSE_RX_PFC is enabled.

Table 22 • Port Signals for Core10GMAC Macro

Name	Width	Direction	Description
RS			
I_CFG_RS_TX_FAULT_EN	1	In	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. The behavior is documented further in the RS section.
I_CFG_RS_TX_FAULT_LOCAL	1	In	The signal informs the RS transmit block to send local fault condition. This signal is normally tied low.
I_CFG_RS_TX_FAULT_REMOTE	1	In	The signal informs the RS transmit block to send remote fault condition. This signal is normally tied low.
I_CFG_RS_TX_IDLE	1	In	The signal informs the RS transmit block to continuously send idle code words. This signal is normally tied low.
I_RS_RX_SRESET	1	In	Synchronous reset for receive RS logic. This signal is normally connected to O_CORE_RX_SRESET.
O_RS_RX_FAULT_LOCAL	1	Out	The signal indicates that the receive RS block is in local fault mode, and if I_CFG_RS_TX_FAULT_EN is asserted the transmit core will start sending the appropriate fault code. The signal is normally connected to O_RS_RX_FAULT_LOCAL from the Rx core. The signal is synchronized into the I_CORE_TX_CLK domain.
O_RS_RX_FAULT_REMOTE	1	Out	The signal indicates that the receive RS block is in remote fault mode, and if I_CFG_RS_TX_FAULT_EN is asserted the transmit core will start sending the appropriate fault code. The signal is normally connected to O_RS_RX_FAULT_REMOTE from the Rx core. The signal is synchronized into the I_CORE_TX_CLK domain.
10GE TX PCS SIGNALS			
O_PMA49_TX_GRBX_SOS	1	Out	Used when interfacing with Microsemi's SerDes with gearbox offload. Only available when TYPE_10G = 0 (that is, 10GBASE-R).
O_PMA49_TX_GRBX_HDR_EN	1	Out	Used when interfacing with Microsemi's SerDes with gearbox offload. Only available when TYPE_10G = 0 (that is, 10GBASE-R).

Table 22 • Port Signals for Core10GMAC Macro

Name	Width	Direction	Description
O_PMA49_TX_GRBX_HDR	4	Out	Used when interfacing with Microsemi's SerDes with gearbox offload. Only available when TYPE_10G = 0 (that is, 10GBASE-R).
O_PMA49_TX_GRBX_DATA_EN	1	Out	Used when interfacing with Microsemi's SerDes with gearbox offload. Only available when TYPE_10G = 0 (that is, 10GBASE-R).
O_PMA49_TX_GRBX_DATA	32 or 64	Out	Used when interfacing with Microsemi's SerDes with gearbox offload. Only available when TYPE_10G = 0 (that is, 10GBASE-R).
I_PCS49_TX_SRESET	1	In	The transmit reset signal. Assert for one or more clock cycles to reset the core.
I_CFG_PCS49_TX_BYPASS_SCRAMBLER	1	In	When asserted high the scrambler is bypassed. Normally tied to 1'b0.
I_CFG_PCS49_TX_TEST_PRBS31_EN	1	In	When asserted high the core is continuously sourcing 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	1	In	When I_CFG_PCS49_TX_TEST_PATTERN_EN is asserted the transmitter will source 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.
I_CFG_PCS49_TX_TEST_PATTERN_TYPE_SEL	1	In	When asserted high the core will transmit a square patter. The square pattern is sourced as 16'hf0f0 towards the PMA. When asserted low the core will source 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	1	In	When asserted high the core uses 64 zeroes as the data-pattern. When asserted low the core uses 64-bit encoding for two Local Fault ordered_sets. See Clause 49 for clarification.

Table 22 • Port Signals for Core10GMAC Macro

Name	Width	Direction	Description
I_CFG_PCS49_TX_TEST_PATTERN_SEED_A	58	In	These values are used for the scrambler seed while running in pseudo-random test-mode. See Clause 49 for clarification.
I_CFG_PCS49_TX_TEST_PATTERN_SEED_B	58	In	These values are used for the scrambler seed while running in pseudo-random test-mode. See Clause 49 for clarification.
10GE RX PCS SIGNALS			
I_PMA49_RX_GRBX_LOCK	1	In	Used when interfacing with Microsemi’s SerDes with gearbox offload. Only available when TYPE_10G = 0 (that is, 10GBASE-R).
I_PMA49_RX_GRBX_SOS	1	In	Used when interfacing with Microsemi’s SerDes with gearbox offload. Only available when TYPE_10G = 0 (that is, 10GBASE-R).
I_PMA49_RX_GRBX_HDR_EN	1	In	Used when interfacing with Microsemi’s SerDes with gearbox offload. Only available when TYPE_10G = 0 (that is, 10GBASE-R).
I_PMA49_RX_GRBX_HDR	4	In	Used when interfacing with Microsemi’s SerDes with gearbox offload. Only available when TYPE_10G = 0 (that is, 10GBASE-R).
I_PMA49_RX_GRBX_DATA_EN	1	In	Used when interfacing with Microsemi’s SerDes with gearbox offload. Only available when TYPE_10G = 0 (that is, 10GBASE-R).
I_PMA49_RX_GRBX_DATA	32	In	Used when interfacing with Microsemi’s SerDes with gearbox offload. Only available when TYPE_10G = 0 (that is, 10GBASE-R).
I_PCS49_RX_SRESET	1	In	The receive reset signal. Assert for one or more clock cycles to reset the core.
I_CFG_PCS49_RX_BYPASS_SCRAMBLER	1	In	When asserted high the scrambler is bypassed. Normally tied to 1’b0.
I_CFG_PCS49_RX_TEST_PRBS31_EN	1	In	When asserted high the core is continuously sourcing 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.

Table 22 • **Port Signals for Core10GMAC Macro**

Name	Width	Direction	Description
I_CFG_PCS49_RX_TEST_PATTERN_EN	1	In	When I_CFG_PCS49_RX_TEST_PATTERN_EN is asserted the transmitter will source 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	1	In	When asserted high the core will transmit a square patter. The square pattern is sourced as 16'hf0f0 towards the PMA. When asserted low the core will source a pseudo-random pattern as determined by: I_CFG_PCS49_RX_TEST_PATTERN_DATA_SEL, I_CFG_PCS49_RX_TEST_PATTERN_SEED_A and I_CFG_PCS49_RX_TEST_PATTERN_SEED_B.
I_CFG_PCS49_RX_TEST_PATTERN_DATA_SEL	1	In	When asserted high the core uses 64 zeroes as the data-pattern. When asserted low the core uses 64-bit encoding for two Local Fault ordered_sets. See Clause 49 for clarification.
10GE PCS RX STATUS & STATS			
O_PCS49_RX_BLOCK_LOCK	1	Out	The signal is asserted when the receiver acquires block delineation.
O_PCS49_RX_HI_BER	1	Out	The signal is asserted when the ber_cnt equals or exceeds 16 indicating a bit error ratio >10–4.
O_PCS49_RX_STATUS	1	Out	This signal indicates that the receiver is in block lock and not in hi_ber state.
O_PCS49_RX_BER_STRB	1	Out	The signal strobes every time 125us_timer_done asserts, that is, every 125us as per the Clause49 specification. The O_PCS49_RX_BER_CNT signal is updated on the same event.
O_PCS49_RX_BER_CNT	8	Out	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 was 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.

Table 22 • Port Signals for Core10GMAC Macro

Name	Width	Direction	Description
O_PCS49_RX_TEST_MODE_ERR_STRB	1	Out	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	8	Out	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 will never assert.
O_PCS49_RX_ERRORED_BLOCK_CNT_STRB	1	Out	When the receiver is in normal mode, this signal strobes each time RX_E state is entered.
APB INTERFACE			
PCLK	1	In	Master clock input
PRESETN	1	In	Active low asynchronous reset
PWRITE	1	In	APB write/read enable, active high
PADDR	16	In	APB address
PSEL	1	In	APB select
PENABLE	1	In	APB enable
PWDATA	8	In	APB data input
PRDATA	8	Out	APB data output
PREADY	1	Out	Ready. The Slave uses this signal to extend an APB transfer.
PSLVERR	1	Out	This signal indicates a transfer failure.
PMA			
O_PMA_TX_RAW_DATA	32	Out	PMA Tx Raw Data. Which clause block data is selected depends on I_CFG_PMA_TX_SEL value. Only available when TYPE_10G = 1 (that is, 10GBASE-KR).
I_PMA_RX_RAW_RDY	1	In	PMA Rx Ready signal. Only available when TYPE_10G = 1 (that is, 10GBASE-KR).
I_PMA_RX_RAW_EN	1	In	PMA Rx enable signal. Only available when TYPE_10G = 1 (that is, 10GBASE-KR).
I_PMA_RX_RAW_DATA	32	In	PMA Rx Raw Data. Only available when TYPE_10G = 1 (that is, 10GBASE-KR).
I_XGMII_RXD	64	In	XGMII Rx Data. Only available when TYPE_10G = 2.
I_XGMII_RXC	8	In	XGMII Rx Control Only available when TYPE_10G = 2.

Table 22 • Port Signals for Core10GMAC Macro

Name	Width	Direction	Description
O_XGMII_TXD	64	Out	XGMII Tx Data. Only available when TYPE_10G = 2.
O_XGMII_TXC	8	Out	XGMII Tx Control. Only available when TYPE_10G = 2.
PCS 73 Gearbox			
I_PCS73_DATA	32 or 64	In	PCS73 Rx Gearbox Data. **Note:** Shim layer data connects to the Rx lane data of PF_XCVR.
I_PCS73_TX_CLK	1	In	PCS73 Tx Gearbox Clock. **Note:** Shim layer Tx clock connects to the Tx lane clock of PF_XCVR.
I_PCS73_RX_CLK	1	In	PCS73 Rx Gearbox Clock. **Note:** Shim layer Rx clock connects to the Rx lane clock of PF_XCVR.
I_PCS73_VAL	1	In	PCS73 Rx Gearbox Valid. **Note:** Shim layer Valid connected to the Rx lane valid of PF_XCVR.

Note: Different combinations of these IO's will be exposed depending on parameter values.

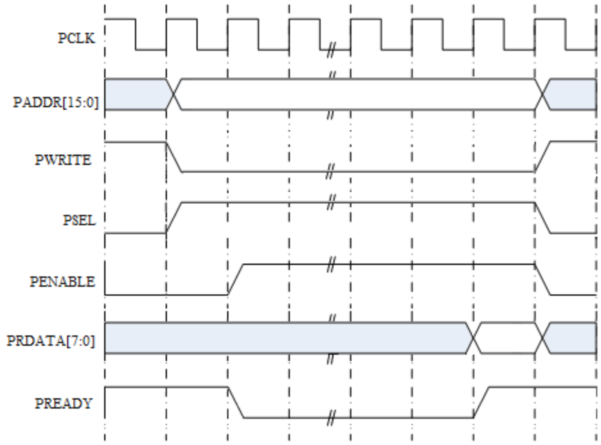
6 Timing Diagrams

6.1 APB Interface

6.1.1 APB Read Timing

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

Figure 8 • **APB Read Timing Diagram**

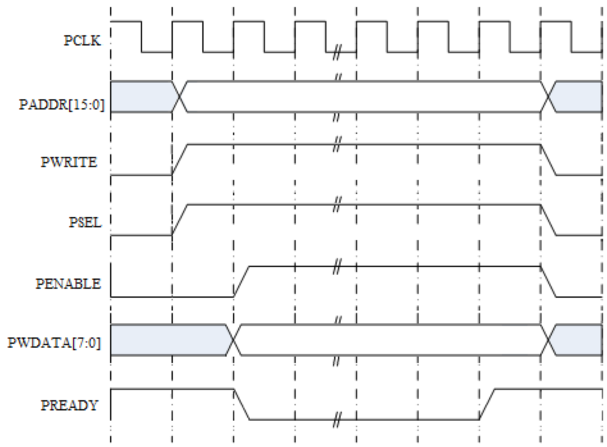


The read transfer starts with the PADDR, PWRITE, and PSEL all changing after the rising edge of PCLK. The first clock cycle of the transfer is called the Setup phase. After the following clock edge the PENABLE is asserted and PREADY deasserts, this indicates that the Access phase is taking place. PADDR, PWRITE, PSEL, and PENABLE all remain valid throughout the Access phase. The transfer completes at the end of the cycle where PREADY asserts, during this cycle PRDATA is valid. PENABLE is deasserted 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.

6.1.2 APB Write Timing

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

Figure 9 • **APB Write Timing Diagram**



The write transfer starts with the PADDR, PWDATA, PWRITE, and PSEL all 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 deasserts, this indicates that the Access phase is taking place. PADDR, PWDATA, PWRITE, PSEL, and PENABLE all remain valid throughout the Access phase.

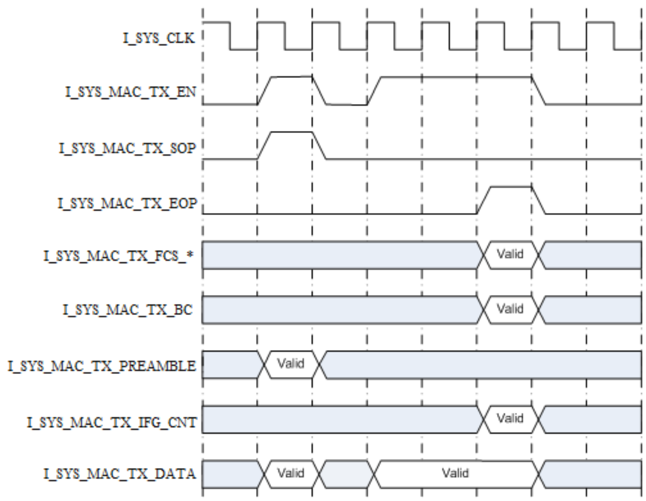
The transfer completes at the end of the cycle where PREADY asserts. PENABLE is deasserted at the end of the transfer. PSEL also goes LOW unless the transfer is to be followed immediately by another transfer to the same peripheral.

6.2 Dataplane

6.2.1 Tx Dataplane Basic Timing

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

Figure 10 • **Tx Dataplane Timing Diagram**



The user initiates a packet transmit event by asserting I_SYS_MAC_TX_EN while also asserting I_SYS_MAC_TX_SOP. The packet transfer ends when the user asserts I_SYS_TX_EOP while also asserting I_SYS_MAC_TX_EN.

- I_SYS_MAC_TX_EN qualifies all signals.
- I_SYS_MAC_TX_SOP qualifies I_SYS_MAC_TX_PREAMBLE.
- I_SYS_MAC_TX_EOP qualifies I_SYS_TX_FCS_*, I_SYS_TX_BC and I_SYS_TX_IFG_CNT.
- 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.

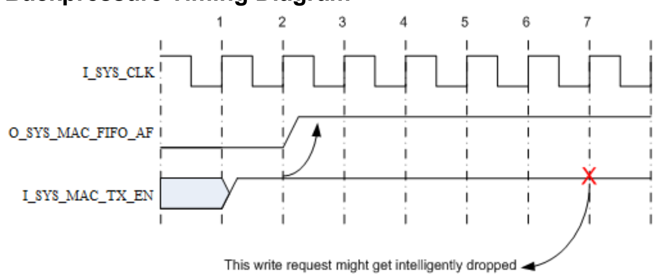
All accesses are full accesses except the I_SYS_MAC_TX_EOP access, where I_SYS_MAC_TX_BC qualifies the number of active bytes.

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

Tx Dataplane Backpressure Timing

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

Figure 11 • **Tx Dataplane Backpressure Timing Diagram**

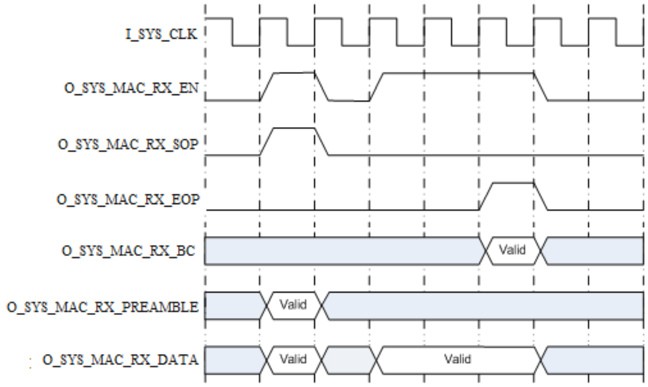


The diagram depicts a situation where the write on the rising edge of cycle '2' causes the almost full flag to assert. When this flag asserts the system transmit FIFO is guaranteed to have space for 4 more writes, so the writes on cycle 3, 4, 5, and 6 are guaranteed to be written correctly, and the write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

6.2.2 Rx Dataplane Basic Timing

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

Figure 12 • **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 asserts while also O_SYS_MAC_RX_EN is 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_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.

6.3 Master Reset

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

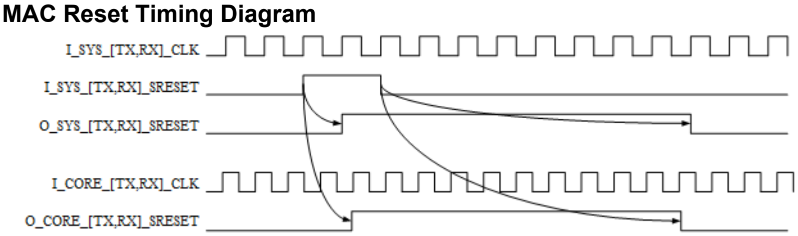
The master reset controller has the following signals:

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

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

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

Figure 13 • **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_SYS_[TX,RX]_SRESET and O_CORE_[TX,RX]_SRESET will remain asserted for a predefined number of clock cycles. Typically around 32-64 clock cycles.

In the transmit direction O_CORE_TX_SRESET is released before O_SYS_TX_SRESET.

In the receive direction O_CORE_RX_SRESET is released after O_SYS_RX_SRESET.

I_SYS_TX_SRESET should be asserted until I_SYS_TX_CLK and I_CORE_TX_CLK are stable.

I_SYS_RX_SRESET should be asserted until I_SYS_RX_CLK and I_CORE_RX_CLK are stable.

7 Tool Flows

7.1 Licensing

Core10GMAC is licensed with evaluation and obfuscated RTL.

7.1.1 Obfuscated

Complete RTL code is provided for the core, enabling the core to be instantiated with SmartDesign. Simulation, Synthesis, and Layout can be performed with Libero software. The RTL code for the core is obfuscated using the IP encryption (encryptP1735.pl) solution.

7.1.2 Evaluation

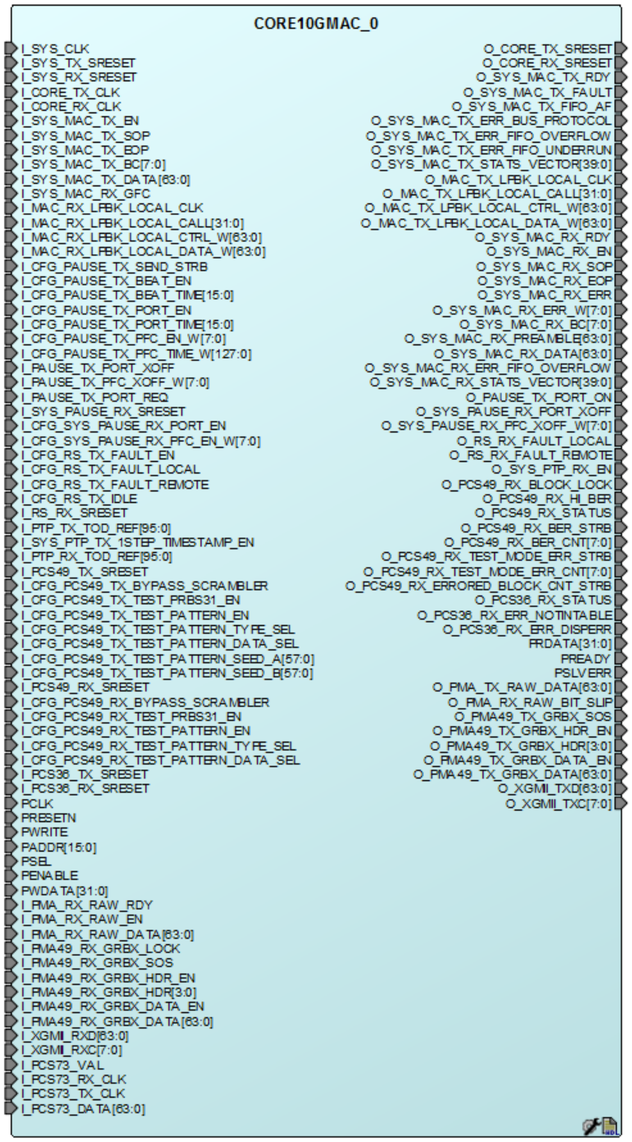
Complete RTL code is provided for the core, enabling the core to be instantiated with SmartDesign. Simulation, Synthesis, and Layout can be performed with Libero software. The RTL code for the core is obfuscated using the IP encryption (encryptP1735.pl) solution and has a time bomb feature which will stop functioning after 4 or 8 hours time at 10Gbps data rate using 64bit at 156.25MHz or 32bit at 312.5MHz clock respectively.

7.2 SmartDesign

Core10GMAC is preinstalled in the SmartDesign IP Deployment design environment.

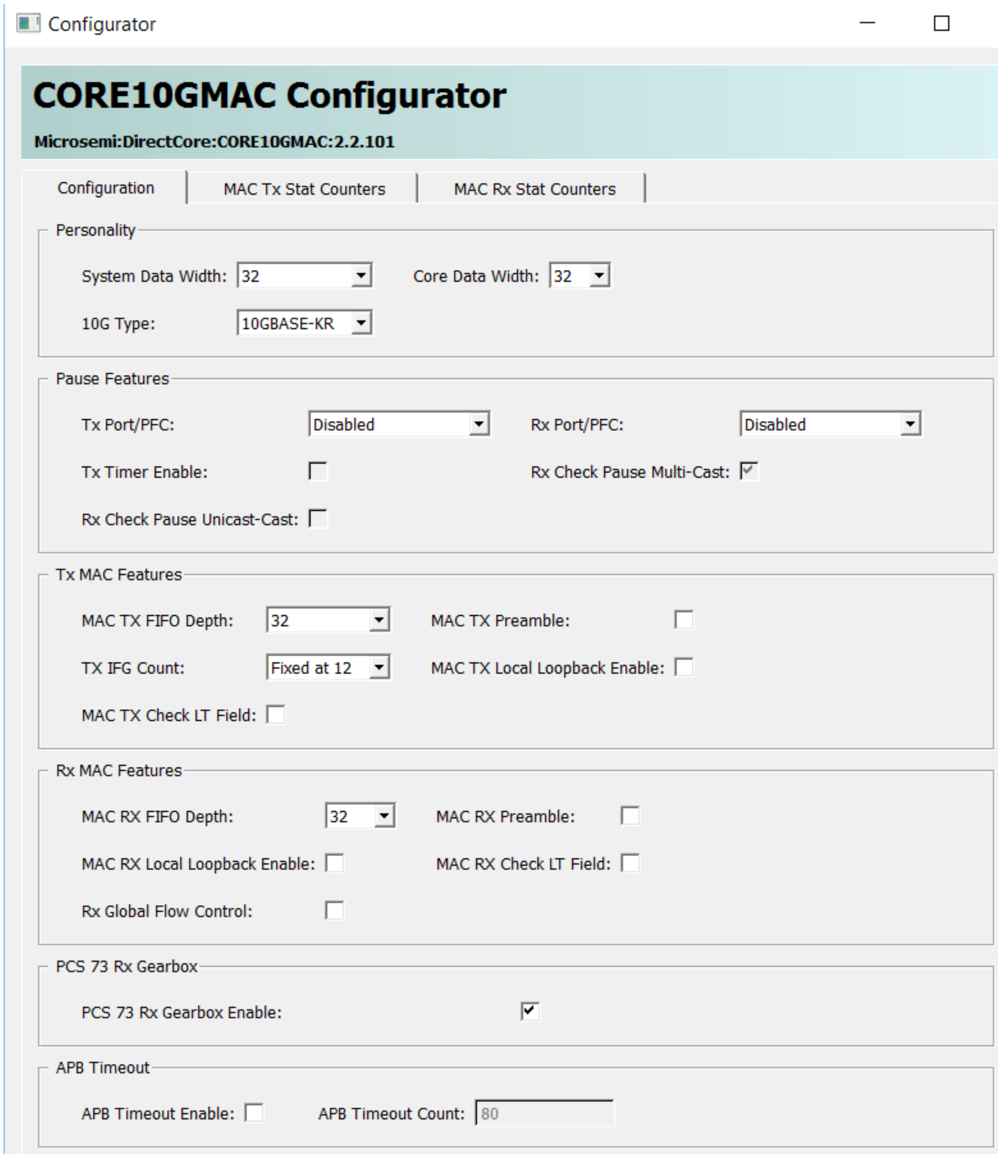
The core should be configured using the configuration GUI within SmartDesign, as shown in the following figure. For Information on using SmartDesign to instantiate and generate cores, see Libero SoC online help.

Figure 14 • Core10GMAC Full I/O View



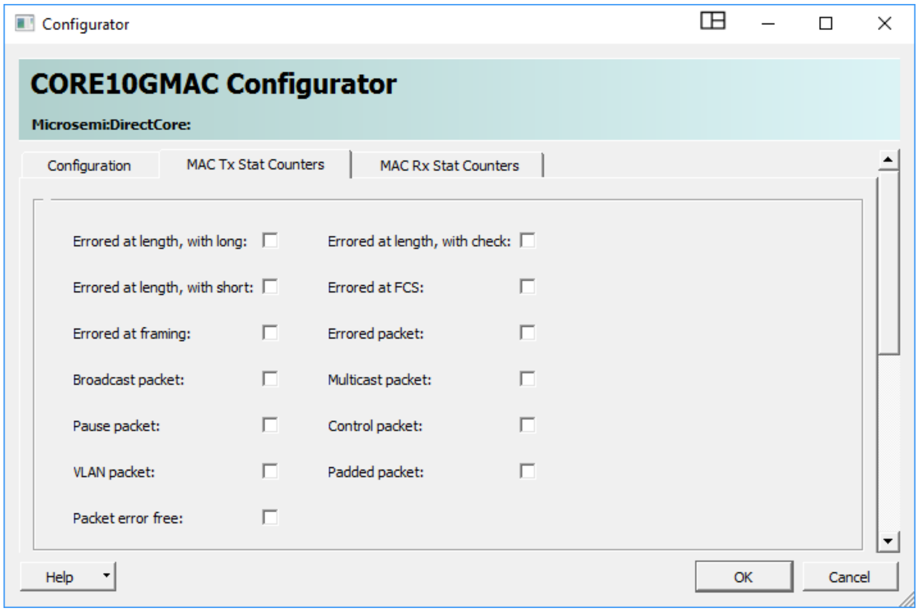
The following figure shows the options available in Configuration tab.

Figure 15 • **Core10GMAC SmartDesign Configuration GUI – Configuration**



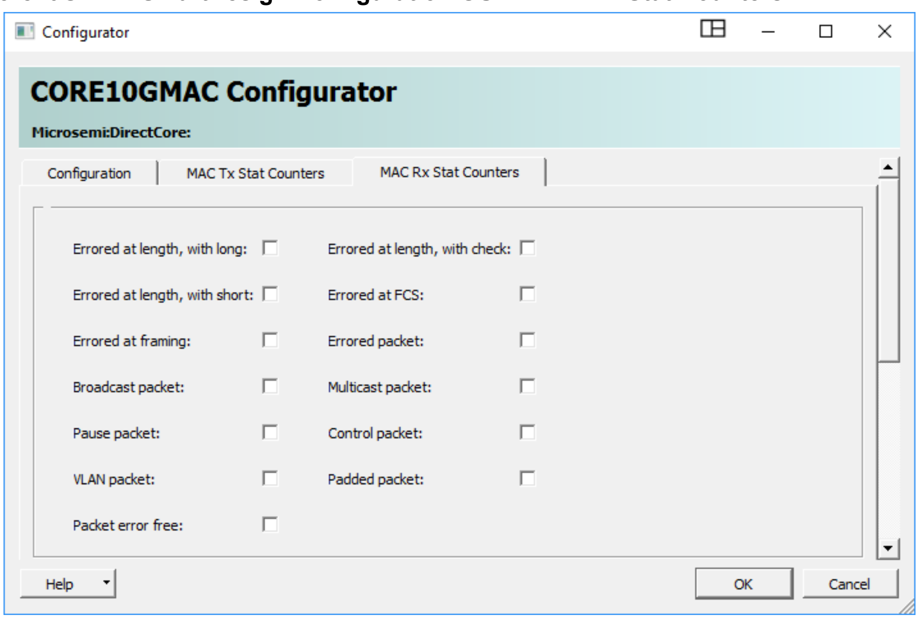
The following figure shows the options available in MAC Tx Stat Counters tab.

Figure 16 • **Core10GMAC SmartDesign Configuration GUI – MAC Tx Stat Counters**



The following figure shows the options available in MAC Rx Stat Counters.

Figure 17 • **Core10GMAC SmartDesign Configuration GUI – MAC Rx Stat Counters**



7.3 Simulation Flows

The user testbench for Core10GMAC is included in all releases.

To run simulations, select the User Testbench flow within the SmartDesign Core10GMAC configuration GUI, right-click the canvas, and select Generate Design.

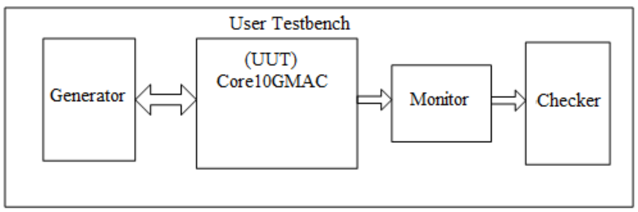
When SmartDesign generates the design files, it will install the user testbench files.

To run the user testbench, Set the design root to the Core10GMAC instantiation in the Libero design hierarchy pane and click the Simulation icon in the Libero SoCPolarFire Design Flow window. This will invoke ModelSim® and automatically run the simulation.

7.3.1 User Testbench

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

Figure 18 • **Core10GMAC User Testbench**



The simulation testbench shown in 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 inputting 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 is not an attempt at an exhaustive testbench.

7.4 Synthesis in Libero SoC PolarFire

After setting the design root appropriately for your design, click the Synthesis icon in Libero SoC. The Synthesis window appears, displaying the Synplicity® project. Set Synplicity to use the Verilog 2001 standard if Verilog is being used. To run Synthesis, click the Run icon.

7.5 Place-and-Route in Libero SoC PolarFire

After setting the design root appropriately for your design, and after running Synthesis, click the Layout icon in Libero SoCPolarFire to invoke Designer. Core10GMAC requires no special place-and-route settings.

7.6 Constraints

As the Core10GMAC is a high-speed IP core, it is important to provide it with constraints in order to Place-and-Route a design without timing violations.

7.6.1 Synthesis Constraints

```
###==== CLOCKS

create_clock -name {I_SYS_CLK} -period 6.2 -waveform {0 3.1 } [ get_ports {
I_SYS_CLK } ]

create_clock -name {I_CORE_RX_CLK} -period 6.2 -waveform {0 3.1 } [ get_ports
{ I_CORE_RX_CLK } ]

create_clock -name {I_CORE_TX_CLK} -period 6.2 -waveform {0 3.1 } [ get_ports
{ I_CORE_TX_CLK } ]
```

8 Ordering Information

8.1 Ordering Codes

Core10GMAC can be ordered through your local Microsemi sales representative. It should be ordered using the following number scheme: Core10GMAC-XX, where XX is listed in Table 23.

Table 23 • **Ordering Codes**

XX	Description
OM	Obfuscated RTL multi-use license