Triple Speed Ethernet MAC

Key Features

- Synthesizable, technology independent VHDL IP Core
- IEEE 802.3-2008 compliant
- Address filter
- 10/100/1000 Media Access Controller
- MII/GMII/RGMII interface
- MDIO Interface
- Automatic preamble generation and removal
- 32-bit CRC generation/verification
- Can be connected to any type of physical layer
- Half Duplex collision check and retransmission
- Payload padding and pad removal
- Half/Full Duplex supported
- Optional AXI4-Lite\Avalon Interface

Supported FPGA

- Any

Supported Simulators

- Active-HDL v9.1 or higher,
- Riviera Pro v2014.02 or higher

Supported Synthesizers

- Mentor
- Xilinx ISE Vivado
- Quartus
- Synopsys Synplify (Lattice and Microsemi versions)

Verification and Validation Flow

- Self-checking test bench
- Static timing analysis
- Code coverage
- Linting

Application

Typical applications: networking

1 OVERVIEW

The Triple Speed Ethernet Media Access Controller (TSE MAC) is a generic triple speed 10/100/1000 Ethernet MAC suited for use in networking applications. In 1000 mode the TSE MAC can be connected with industry standard PHY devices using RGMII or GMII interface. When using 10/100 mode the TSE MAC uses the MII interface.

This IP core is designed to interface with both AXI4-Lite and Avalon protocol, or as a standalone core.

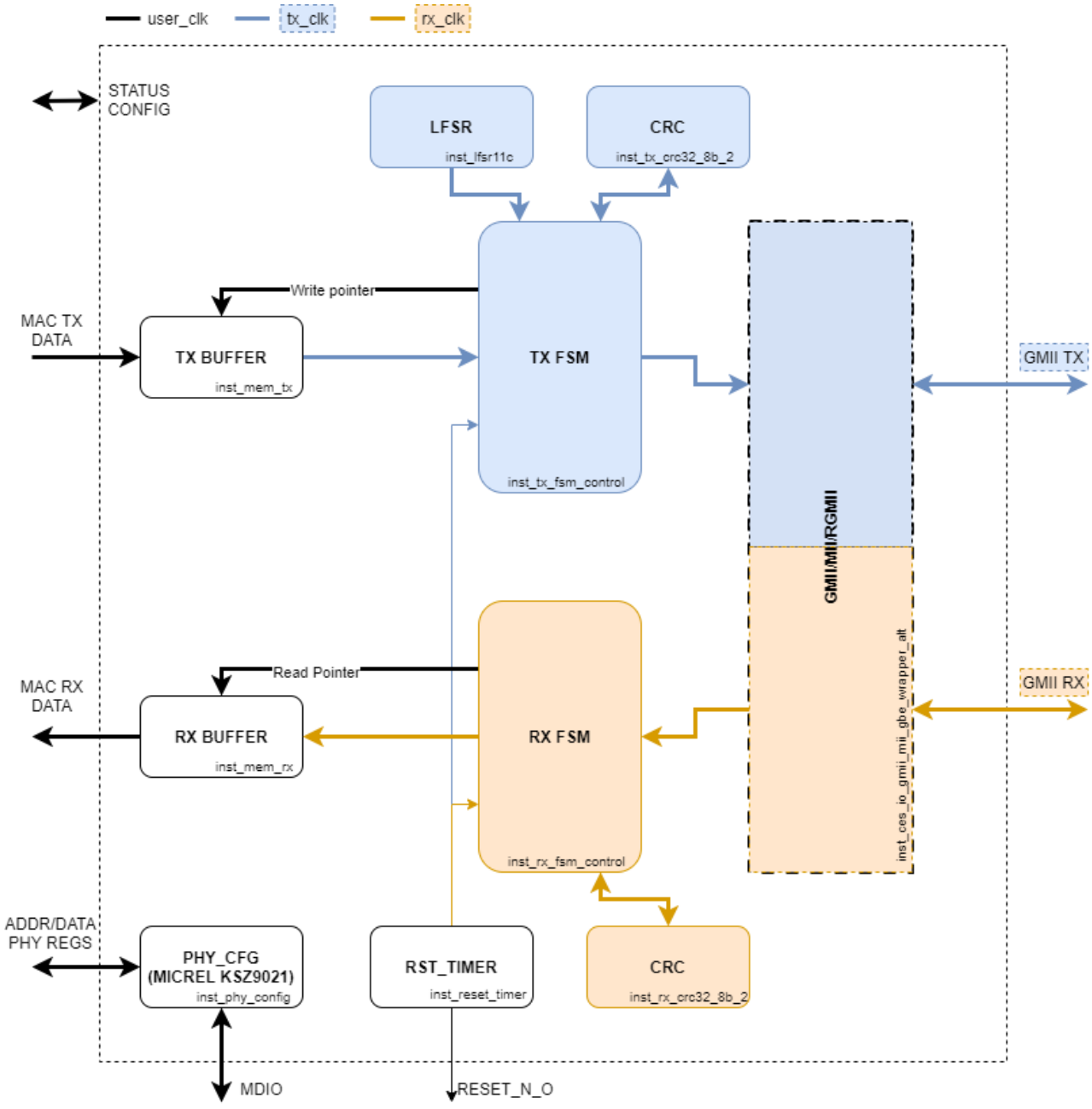

Figure 1. Block diagram of the TSE MAC module

The Core is capable to produce a standard (as per IEEE 802.3) ethernet frame, with the most-significant octet (byte) first; within each octet the least-significant bit is transmitted first.

Preamble	Start Frame Delimiter	MAC Destination	MAC Source	802.1Q Tag	Length/Type	Payload	CRC
7B	1B	6B	6B	4B	4B	46-1500B[1]	4B

1.1 LICENSING AND ORDERING INFORMATION

Information about licensing, pricing and availability is available on the company website (http://www.campera-es.com/), or after contacting the company or your local distributor. Three types of licenses keys may be available:

- Netlist, device locked
- Source code: project based
- Source code: site based

Evaluation licenses can be requested

- Simulation Only License
- Full System Hardware Evaluation License

1.2 DESIGN METHODOLOGY

The code is compiled following an internal coding standard, and the flow is developed under constant simulation check. The working code undergoes a linting procedure using Aldec Alint software to spot main code errors or generic issues.

Each piece of design is simulated in a dedicate testbench, possibly of self-checking type. Various combinations of generic parameters are tested, looking for critical situations. The whole span of admitted values is explored, and care is taken about treatment of values which are not representable or leading to standard errors.

Synthesis is tested on selected sample targets, to collect data about resource occupation and operational frequencies.

NAMING CONVENTIONS:

_e	one-CLK early sample
_d	one-CLK delayed sample
_d2	two-CLKs delayed sample
_n	active low signal
C_	constant
s_	signal
_i	input port
_o	output port
_io	inout port

[1] The Core supports also Jumbo frames, with a larger payload

t_	Type
_st	FSM state

2 PRODUCT SPECIFICATION

The CES_COMM_TSE_MAC module is part of the CES COMMUNICATION Library, and is available in several configurations:

- Simple Interface or AXI/Avalon register mapped interface
- FPGA Vendor and Family. The DDR feature of RGMII and the clock multiplexing require hard primitives, FPGA vendor and family must be communicated before ordering
- PHY Interface: RGMII or GMII/MII
- External PHY model (Micrel KSZ9021, Micrel KSZ9031 and Marvell Alaska are HW proven, but the core is designed to be independent of the PHY)

3 INTERFACE

Given the high number of possible configurations of the core, and hence different interfaces, we will describe the Full Duplex Programmable Core.

3.1 GENERIC DESCRIPTION

Table 1 report the list of generics:

- Generic Name: this column identifies the generic name and width, in case of std_logic_vector. As a rule, a name followed by [n-1:0] indicates a std_logic_vector bus of width n.
- Description

Table 1 – generic descriptions

Generic	Description
g_phy_addr [4:0]	Set the address of the PHY device
g_clk_frequency	User clock frequency (at least 125 MHz)
g_buffer_size	TX/RX buffer size
g_simulation	0: normal operation, 1: assert the link status for simulation purposes

3.2 PORT DESCRIPTION

All ports are std_logic or std_logic_vectors, as a rule, a name followed by [n-1:0] indicates a std_logic_vector bus of width n.

Input ports are indicated with a *_i* suffix, whether output ports are indicated with a *_o* suffixe-

Table 2 – ports description

Port name	Description

user_clk_i	User clock input. Must be > 125Mhz for 1Gb operations. The user clock must always be present
user_rst_n_i	Synchronous reset, user_clk_i domain, active low
config_reg_i [31:0]	MAC Configuration Register - bit 0: (1) Automatic padding of short frames. Requires that auto-CRC insertion be enabled too. (0) Skip padding. User is responsible for adding padding to meet the minimum 60-byte frame size - bit 1: (1) Automatic appending of 32-bit CRC at the end of the frame (0) Skip CRC32 insertion. User is responsible for including the frame check sequence - bit 2: (1) promiscuous mode enabled (0) disabled, i.e., destination address is verified for each incoming packet - bit 3: (1) accept broadcast rx packets (0) reject - bit 4: (1) accept multi-cast rx packets (0) reject - bit 5: (1) filter out the 4-byte CRC field (0) pass along the CRC field. - bit [7:6]: Speed, (00) = force 10 Mbps, (01) = force 100 Mbps, (10) = force 1000 Mbps, (11) = auto-negotiation (default) - bit 8: (1) = full-duplex (default), (0) = half-duplex - bit [10:9]: test mode (00) = normal mode (default), (01) = loopback mode (at the phy), (10) = remote loopback, (11) = led test mode - bit 11: software power down mode. (1) = enabled, (0) = disabled (default). - bit [27:12]: 16-bit clock skew for TX/RX clock. The register and number of bits per clock are PHY dependent -- bit 28: phy reset - bit [31:29]: spare
mac_addr_i [47:0]	This network node 48-bit MAC address. The receiver checks incoming packets for a match between - the destination address field and this MAC address. - The user is responsible for selecting a unique hardware address for each instantiation. - Natural bit order: enter x0123456789ab for the MAC address 01:23:45:67:89:ab
phy_config_change_i	Configuration signals are synchronous with the user-side CLK. - optional pulse to activate any configuration change below. - Not needed if the default values are acceptable. - Ignored if sent during the initial PHY reset (10ms after power up)
sreg_read_start_i	1CLK wide pulse to start read transaction. Will be ignored if the previous (or write) transaction is yet to be completed.
sreg_write_start_i	1CLK wide pulse to start read transaction. Will be ignored if the previous (or read) transaction is yet to be completed.

sreg_regad_i [15:0]	32 register address space for the PHY (ieee 802.3) - 0 - 15 are standard PHY registers as per IEEE specification. - 16 - 31 are vendor-specific registers - 256+ are extended registers
sreg_is_ext_i	1: extended register, 0: standard register
sreg_data_o [15:0]	output data read from PHY register
sreg_dv_o	output data valid, mark valid sreg_data_o read data
sreg_data_i [15:0]	Input data to write to the PHY register
mac_tx_data_i [7:0]	MAC reads the data at the rising edge of user_clk_i when mac_tx_data_valid_i = '1'
mac_tx_data_valid_i	TX data valid
mac_tx_eof_i	'1' when sending the last byte in a packet to be transmitted. Aligned with mac_tx_data_valid_i
mac_tx_cts_o	MAC-generated Clear To Send flow control signal, indicating room in the - tx elastic buffer for a complete maximum size frame 1518B. The user should check that this signal is high before deciding to send the next frame.
mac_rx_data_o [7:0]	USER reads the data at the rising edge of user_clk_i when mac_rx_data_valid_o = '1'
mac_rx_data_valid_o	RX output data valid
mac_rx_sof_o	'1' when sending the first byte in a received packet. Aligned with mac_rx_data_valid_o
mac_rx_eof_o	'1' when sending the last byte in a received packet. Aligned with mac_rx_data_valid_o
mac_rx_cts_i	User-generated Clear To Send flow control signal. The receive MAC checks that this signal is high before sending the next mac_rx_data_o byte.
reset_n_o	PHY hard reset, active low. It is generated with the user_clk_i clock. If the user_clk_i is coming from the PHY the user has to provide the PHY reset_n outside of this module with another clock
mclk_o	MDIO clock (1MHz)
mdio_io	(tri-state) serial interface
mii_tx_clk_i	MII tx clock from PHY. Continuous clock. (10/100 Mbps only) - 25 MHz (100 Mbps), or 2.5 MHz (10 Mbps) depending on speed - accuracy: +/- 100ppm (MII) - duty cycle between 35% and 65% inclusive (MII).
gmii_tx_clk_o	GMII tx clock to PHY. Continuous clock. 125MHz (1000 Mbps only)

gmii_mii_txd_o [7:0]	GMII TX data, in MII only bits (3:0) are used. tx data (when TX_EN = '1' and TX_ER = '0') or special codes otherwise (carrier extend, carrier extend error, transmit error propagation). See 802.3 table 35-1 for definitions.
gmii_mii_tx_en_o	GMII TX enable
gmii_mii_tx_er_o	GMII TX error
gmii_mii_crs_i	carrier sense
gmii_mii_col_i	collision detected
gmii_mii_rx_clk_i	continuous receive reference clock recovered by the PHY from the received signal - 125/25/2.5 MHz +/- 50 ppm. - Duty cycle better than 35%/65% (MII) - 125 MHz must be delayed by 1.5 to 2.1 ns to prevent glitches
gmii_mii_rxd_i [7:0]	rx data. 8-bit when 1000 Mbps. 4-bit nibble (3:0) when 10/100 Mbps.
gmii_mii_rx_dv_i	RX data valid
gmii_mii_rx_er_i	RX data error
speed_status_o	RXC clock speed, 00 = 2.5 MHz, 01 = 25 MHz, 10 = 125 MHz, 11 = no RXC clock from the PHY
link_status_o	0 = link down, 1 = link up
duplex_status_o	0 = half duplex, 1 = full duplex
n_rx_frames_o[15:0]	number of received frames
n_rx_bad_crcs_o [15:0]	number of BAD CRCs among the received frames
n_rx_frames_too_short_o [15:0]	number of rx frames too short (<64B)
n_rx_frames_too_long_o [15:0]	number of rx frames too long (>1518B) when jumbo frames are disabled
n_rx_wrong_addr_o [15:0]	number of rx frames where address does not match (and promiscuous mode is off)
n_rx_length_errors_o [15:0]	number of rx frames with length field inconsistent with actual rx frame length

4 GENERAL DESCRIPTION

The block diagram in Figure 1 shows the modules that constitute the MAC, each of the module will be briefly described in the next chapters.

The TX buffer (inst_mem_tx) and RX buffer (inst_mem_rx) modules are dual port, dual clock memories, used as buffer to/from the user clock domain. The size of the buffers is 2*g_buffer size, so they can hold in memory 2 frames if the generic g_buffer size would be set to the maximum frame size.

The Triple Speed Ethernet MAC uses modules from other Campera-ES VHDL libraries:

- ces_util_lib: RAM module and LFSR
- ces_io_lib: CRC parallel calculator

The user shall include the files from the ces_util_lib library and compile them in a library called ces_util_lib (both in simulation and synthesis), and the ces_io_crc_par.vhd module must be compiled in a library called ces_io_lib.

4.1 TX FSM (inst_tx_fsm_control)

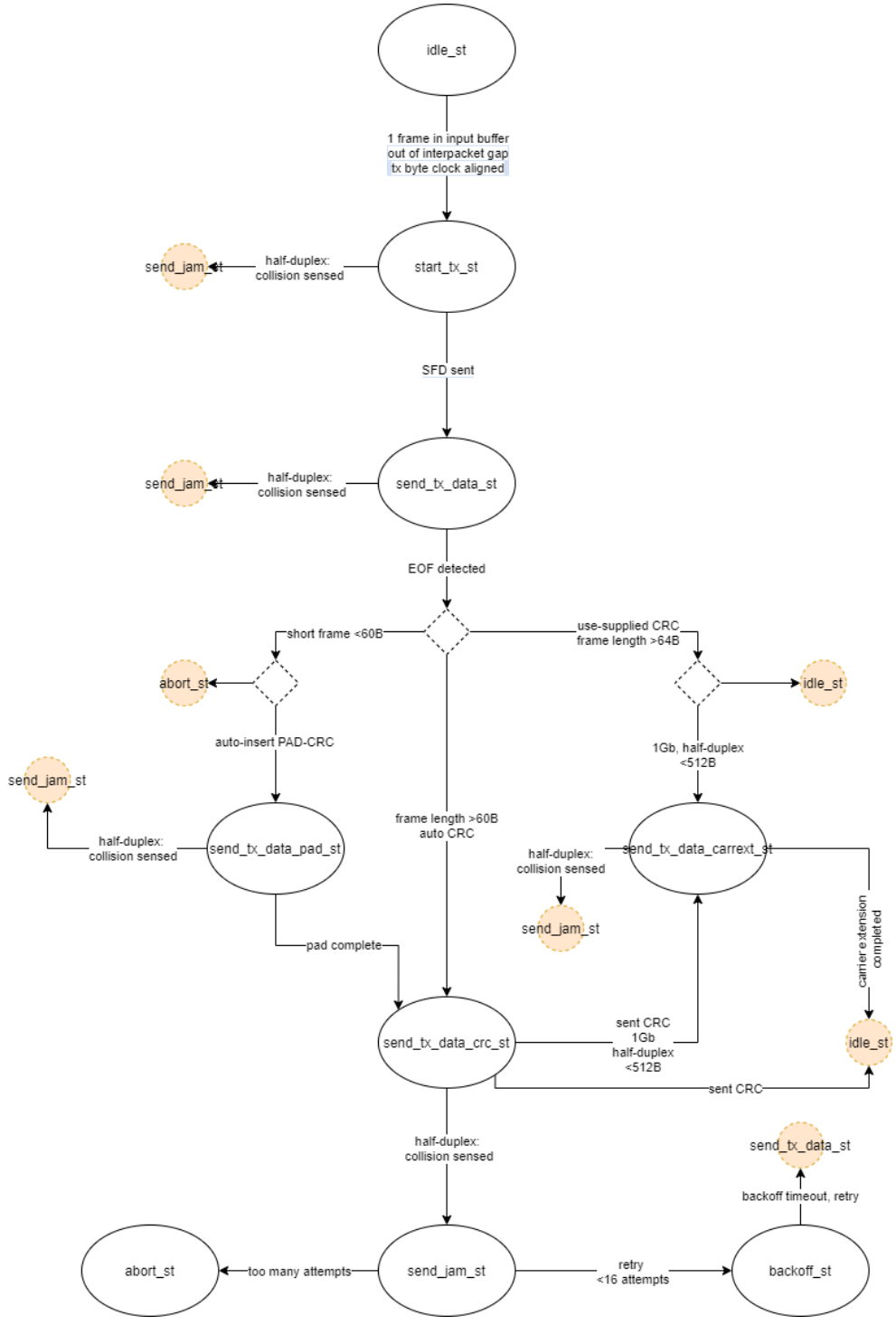

Figure 2 – TX Finite state machine

The TX_FSM module is responsible to capture data form the TX buffer and prepare the ethernet frame for the GMII/RGMII module.

The module:

- Generates the write pointer for the TX buffer, incrementing the pointer with mac_tx_data_valid_i
- Generates the read pointer for the TX buffer, while in send_tx_data_st
- Resync the pointers in the tx clock domain, and computer the free space in the buffer (s_mac_tx_buf_size)
- Generates the TX CTS (Clear To Send) handshake signal, checking the free space in the TX buffer. CTS is asserted when at least half of the TX buffer is free and no retransmission after collision is running
- Generates the Inter Packet Gap counter
- Generates the retry backoff time, depending on the retry attempt number. The random numbers are read form the LFSR external module
- Generates the preamble in start_tx_st
- Flip the input CRC and reorder the CRC to be included in the ethernet frame
- Mux the preamble, payload, padding, CRC, and carrier extend to create the ethernet frame

4.2 RX_FSM (inst_rx_fsm_control)

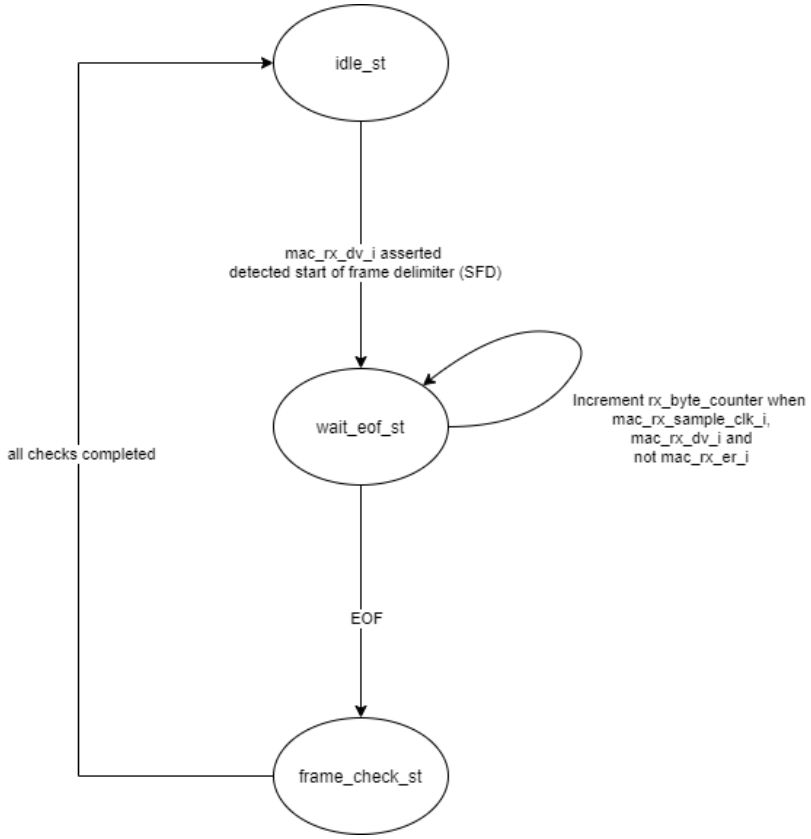

Figure 3 - RX Finite State Machine

The RX_FSM module is responsible to capture data from the GMII/RGMII module, check the frame validity and fill the RX buffer.

The module:

- Check if the frame is too short, also incrementing a diagnostic counter
- Check for valid destination address: the mac_rx_config_i bits are used to enable promiscuous mode, broadcast, and multicast addresses

- Check the length type field, to extract the packet length
- Check for length errors (short frames or too long frames) raising an error bit and increasing a diagnostic counter
- Check the received frame and mark it as valid if
  - CRC good
  - Length good (>=64B and <1518B)
  - Valid address
- Filter out the CRC when the configuration bit is set accordingly
- Generates the write pointer for the RX buffer
- Generates the read pointer for the RX buffer
- Computes the free space in the RX buffer and reconstruct the EOF

4.3 RST_TIMER (inst_reset_timer)

This module:

- Generates a 10ms active low reset_n_o pulse at power up or after a user reset trigger. for the PHY If the g_simulation generic is set the reset is 100 us, to allow for faster simulation startup
- Generate a pulse 90ms after reset to configure the PHY (50 us after reset when g_simulation is asserted)
- generates synchronous reset in the RX CLK and TX CLK domains

4.4 PHY_CFG (inst_phy_config)

This module:

- Reads and writes PHY registers through MDIO interface
- MII/GMII: see 802.3 22.2.4 Management functions
- Supported registers (can be extended and/or read/write by user commands) reg0 control, reg1 status and reg2 phy ID

The module has 2 methods to configure the PHY:

- At power up or when the config_change_i input is asserted a write/read sequence is started. The following parameters are read at the module input
  - ✓ phy_reset_i
  - ✓ speed_i
  - ✓ duplex_i
  - ✓ power_down_i
  - ✓ clk_skew_i
- Each PHY register can be accessed read/write through a standard user interface (address, read/write, rx data tx data). This can be used also for debug purposes,

At the end of each transaction on the MDIO link a s_mi_transaction_complete is asserted, and the FSM capture that signal and starts a new read transaction. That way a loop is done, and every 256 clock cycles a s_mi_transaction_complete → s_mi_read_start sequence is generated, for continuous status register reading.

4.4.1 MDIO interface

The PHY is managed through a two-wire Management Data Input/Output (MDIO) interface. The MDIO is a synchronous serial link comprising two signals: a clock MCLK and a bi-directional data line MDIO. The MDIO timing diagram is specified by the IEEE 802.3 standard.

The component ces_comm_tse_mac_mdio.vhd implements a read or write transaction, converting a 16-bit parallel register value and 5-bit register address to a serial stream and vice-versa.

A write transaction (sending 0xA50F to register address 10) is illustrated below:

The write sequence consists of

(a) 32-bit preamble field (all ones)

(b) "0101"

(c) 5-bit PHY address, here fixed at 0

(d) REG address MI_REGAD[4:0]

(e) "10"

(f) 16-bit data field MI_TX_DATA[15:0]

A read transaction (receiving 0Xffff from register address 10) is shown below:

The read sequence consists of

(g) a 32-bit preamble field (all ones) to the PHY

(h) "0110" to the PHY

(i) 5-bit PHY address, here fixed at 0 to the PHY

(j) REG address MI_REGAD[4:0] to the PHY

(k) "Z"

(l) "0" from the PHY

(m) 16-bit data field MI_RX_DATA[15:0] from the PHY

The PHY generally sets a speed limit for the MDIO interface. For example, the Micrel KSZ9021 PHY defines the typical MCLK frequency as 2.5 MHz. The ces_comm_tse_mac_mdio.vhd component includes a constant (C_MCLK_COUNTER_DIV) to adjust the MCLK frequency by dividing the processing user clock. The designer should adjust this constant prior to synthesis, depending on the specific MDIO timing requirements of the PHY IC.

The default MDIO clock frequency is set to 1 MHz

4.5 PHY INTERFACE

The PHY Interface module is the RGMII or GMII interface module. The instantiated module depends on the FPGA target because it uses low level primitives such as clock muxes and IODELAY for MII/GMII interface and Double Data Rate IO for RGMII interface.

The IP is provided with the proper PHY interface module, customized for the FPGA target. Most FPGA families of the same vendor share the same low-level primitives and can be instantiated in VHDL, but customization might be needed for some family.

Care must be used when placing I/O pins not to introduce delays or skew among clock groups. It is suggested to place the input/output pin/registers in the IOB Pad.

The general strategy is to use the extended registers of the PHY to introduce a skew on the TX/RX clocks. It is usually not necessary to delay data and controls as long as PCB lines' lengths are the same.

4.5.1 MII/GMII (inst_ces_io_gmii_mii_gbe_wrapper_alt)

The component reformat signals exchanged between the FPGA and an external GMII/MII PHY.

The receive clock RX_CLK is always supplied by the PHY to the FPGA. Its frequency (2.5,25 or 125 MHz) depends on the 10/100/1000 Mbps link speed negotiated by the PHY.

The transmit clock direction depends on the selected speed:

The MII TX_CLK is from the PHY to the FPGA, at speeds of 2.5 or 25 MHz (10/100 Mbps link)

The GMII GTX_CLK is a 125 MHz from the FPGA to the PHY (1000 Mbps link) and is connected to the received RX_CLK from the PHY at 125 MHz

The module is vendor and family dependent as the glitch free clock mux is conveniently taken from the vendor libraries. If the FPGA vendor or the family is not included in the supported device's list, please ask your reference at Campera-ES.

Please beware that the pin placement can affect the use of the proper primitive and the fitter could raise errors. As an example, in an Intel Cyclone 10 the clock mux has specific connections to the device pins. If they are not mapped correctly the fitter will raise an error. In our case we have two clocks to feed the TX part.

The mii_tx_clk_i is an input pin from the PHY at 100 Mbps and 10 Mbps and the gmii_mii_rx_clk_i 125 Mhz which is used from the MAC to send the clock at the PHY in 1G mode

The Cylone 10 ALTCLKCTRL module require that the input pin must be placed

- inclk[0] CLK_[2,3][A..L]_0p or any counters from adjacent I/O PLLs.
- inclk[1] CLK_[2,3][A..L]_0n or any counters from adjacent I/O PLLs.
- inclk[2] CLK_[2,3][A..L]_1p or any counters from adjacent I/O PLLs.
- inclk[3] CLK_[2,3][A..L]_1n or any counters from adjacent I/O PLLs.

notice that the inclk(0), which is connected to the mii_tx_clk_i input, MUST come from a 0p clock PIN, inclk(1), which is connected to the gmii_mii_rx_clk_i input, from a 0n clock PIN.

The Xilinx GMII/MII module uses a BUFGMUX primitive

4.5.2 RGMII

The component reformat signals exchanged between the FPGA and an external RGMII PHY.

The RGMII component translates the 6-pin receive PHY interface (DDR, 4-bit nibblets) into a simpler (single clock edge, data-byte) interface.

The RGMII is vendor and family dependent, as it uses Double Data Rate registers to convert the incoming data into an 8-bit bus and to convert an 8-bit bus into an outgoing 4 bit DDR bus. The DDR module are usually vendor/family dependent so please specify your FPGA model before ordering.

Its good practice to delay the clock with respect to data to have a glitch free, well-sampled data bus. The baseline method is to set the delay in the proper extended control register of the PHY. This method works well with Micrel KSZ9021 PHY but may not be supported by other Manufacturers' PHYs. An alternative method using delay element in the FPGA is also coded but currently commented out. Again, please specify your FPGA model before ordering

Figure 4 shows the RX chain of the MAC RGMII module

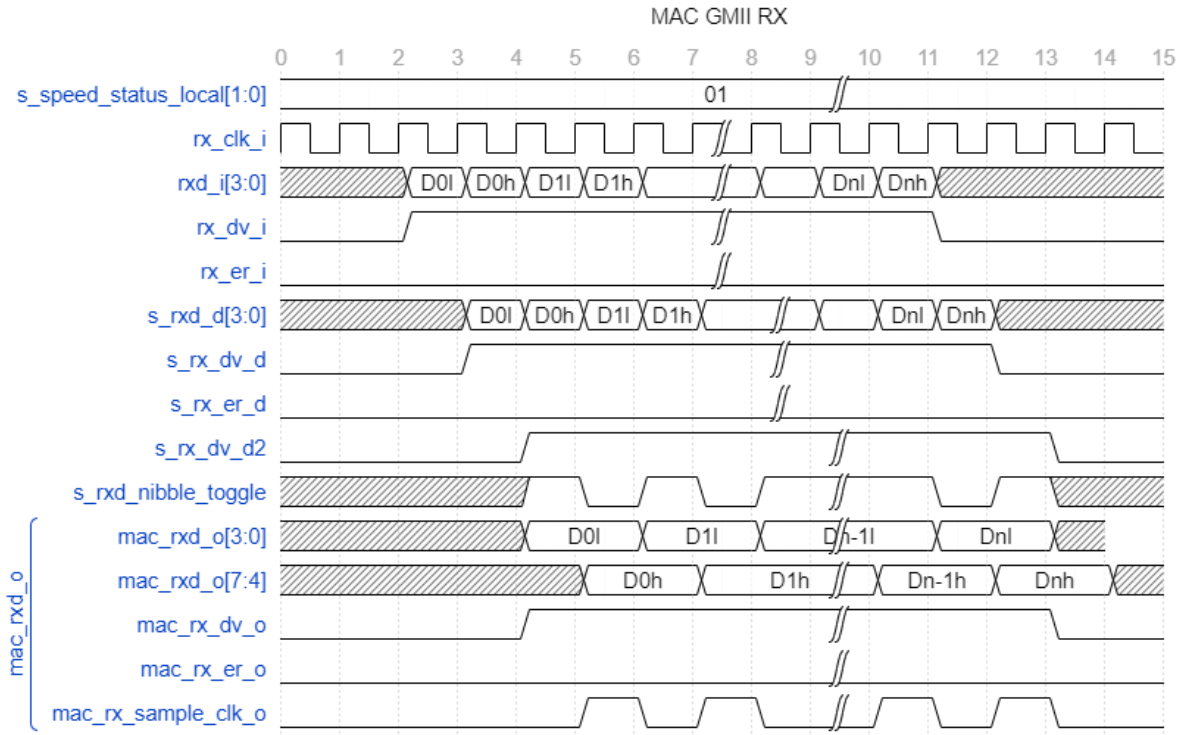

Figure 4 - MAC GMII RX timing diagram

4.6 CRC

All 802.11 frame include a 32-bit CRC (frame check sequence FCS). As part of the MAC layer processing, the module automatically appends the 32-bit CRC at the end of the transmitted frame and verifies the CRC validity on received frames from the PHY.

The module ces_io_.crc_par.vhd is a general-purpose parallel module that computes CRC and is used for both CRC32 generation and CRC check.

The module is configured as follows:

(a) The init value is set to x"FFFFFFFF" at the start of frame.

(b) one-byte input data, 32-bit output data, parallel implementation of the CRC32 takes one clock for each input byte.

(c) The generator polynomial is

$G(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1.$

(d) The CRC is flipped and inverted prior to transmission (LSB first)

(e) The entire received frame (including the 32-bit CRC at the end) is fed into the component. When the frame is error free, the resulting CRC is the residual value 0xC704DD7B.

4.7 LFSR (inst_lfsr11c)

The xorshift random number generator produces a sequence of 2^g_data_w -1 by making a xor of a computer word with a shifted version of itself.

Computing such xorshift opeations for various shifts and arguments provides extremely fast and simple RNGs.

The LFSR is used as pseudo random number generator during retransmission to reduce the probability of collisions, see 802.3 standard section 4.2.3.2.5 for details.

5 FUNCTIONAL TIMING

The user is responsible to feed data to the MAC FIFO-like input interface.

1. Check if the TX_CTS signal is asserted (at least half of the input buffer free)
2. Write data and data valid in the user clock domain, assert the EOF signal for one clock cycle to mark the last byte of the ethernet data frame

The TX state machine starts to read data from the input buffer when an EOF is received, forms the ethernet frame with preamble MAC addresses, fields, data and CRC and send them to the PHY interface (GMII/MII/RGMII depending on the configuration).

Data are sent to the PHY with the chosen interface (see TX/RX RGMII interface in Figure 5) and received with the same interface type.

The RX state machine tries to find the preamble and SFD byte in the input data, then waits for an EOF and check that the following conditions are true:

- Valid CRC
- Not too short frame (<64B)
- Not too long frame (>1518B)
- Valid address received
- Frame length consistent with the type/length field

The received data are written to the output buffer with or without the 4-byte CRC, as specified by the proper control register bit, and finally the write pointer is confirmed (if there was an error the pointer would have been set back to the original position, discarding the RX frame).

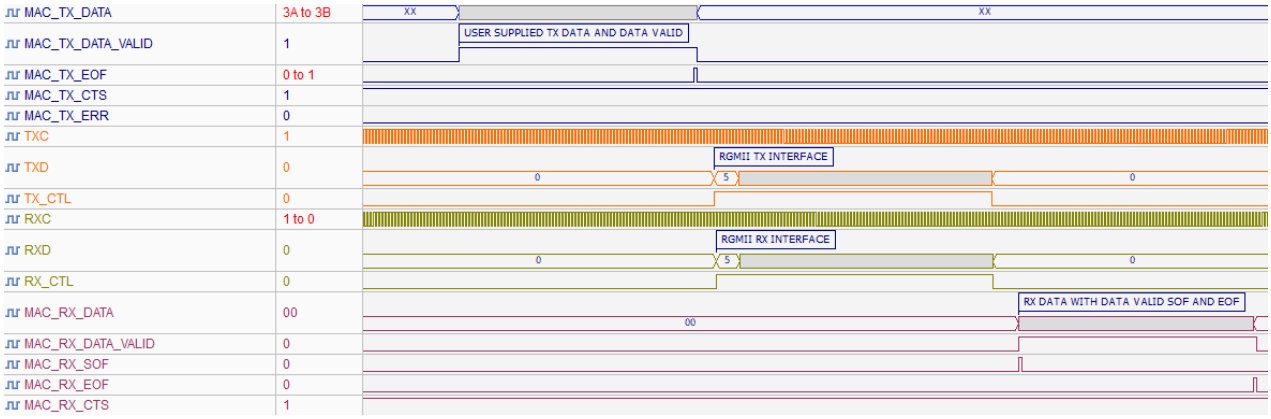

Figure 5 – RGMII interface

6 SIMULATION

The TSE MAC core is verified with a VHDL testbench, emulating the PHY as a simple loopback.

The following test are executed with the testbench and automatically checked:

- Test of minimum (64B) frame size: 1Gbps
- Test of less than minimum frame size: 1Gbps
- Test of 1500 bytes standard frame size: 1Gbps
- Test broadcast: 1Gbps
- Test Jumbo frame: 1Gbps
- Test of 1500 bytes standard frame size: 100Mbps
- Test of 1500 bytes standard frame size: 10Mbps

To run the simulation a script is provided in the *macros* folder, which can be run with the command:

vsimsa -do compile.do

The command requires Active-HDL to be installed and in the path.

The relevant source files are compiled and the testbench is run, the log shall report all test as "PASSED"

7 SYNTHESIS

The following tables show some synthesis results. Here examples are given of the resource utilization in selected FPGAs.

Please ask your local distributor or contact us if you need resource usage for other devices.

Family Device	Fmax (MHz)	LCs	Registers	Memory	Mult/DSP
Intel Cyclone 10	268	655 (ALMs)	893	294912 (Memory Bits)	0
Xilinx Kintex 7	311	700 (LUTs)	880	8 BRAMs	0

Table 3. TSE MAC parameters: RGMII, buffer size 8192B

8 PRODUCT SUPPORT

3 months maintenance with

- Delivery of the IP Core and documentation updates, minor and major versions changes
- Email support

9 REVISION HISTORY

REV.	DESCRIPTION	DATE
1.0	Initial release	12/9/2019