GbE -10 GbE STACK TCP/UDP/IP/PING/ARP

Key Features

- Synthesizable, technology independent VHDL IP Core
- IEEE 802.3
- Ethernet packet encapsulation (RFC 894), IPv4
- Up to 32 UDP TX/RX channels
- Run time programmable IP addresses and Gateway, source and destination ports and MAC address
- Supported protocols: IPv4 (no fragmentation), ARP (with Cache), IMCP (Ping), UDP/IP Unicast and Multicast, TCP
- The core is Ethernet MAC-independent but it seamlessly integrates with Campera-ES TSE MAC Core.
- No external processor required
- Optional AXI4 Lite Interface

Supported FPGA

- Any

Supported Simulators

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

Supported Synthesizers

- Mentor Precision r2013b.15 or later,
- Xilinx ISE 14.7/Vivado 2013.4 or later,
- Quartus II 14.1 or later,
- Synopsys Synplify 2015.03 or later (Lattice and Microsemi versions)

Verification and Validation Flow

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

Application

Typical applications include: Networking

1 OVERVIEW

The ces_comm_eth_stack is a generic Internet protocol stack designed to support 1Gbps-10Gbps throughputs on low-cost FPGAs.

Its goal is to achieve the highest theoretical throughput achievable for a given medium.

The following protocols are implemented in modular VHDL components:

- TCP server/client (optional)
- UDP
- ARP
- PING.

Auxiliary components for streaming, test signal generation, and bit error rate measurement are also included. These components can be used in the application as needed.

The code interfaces seamlessly with the CES_COMM_TSE_MAC Triple Speed Ethernet Media Access Controller. The MAC interface is generic and simple enough to interface with any Ethernet MAC component with minimum glue logic.

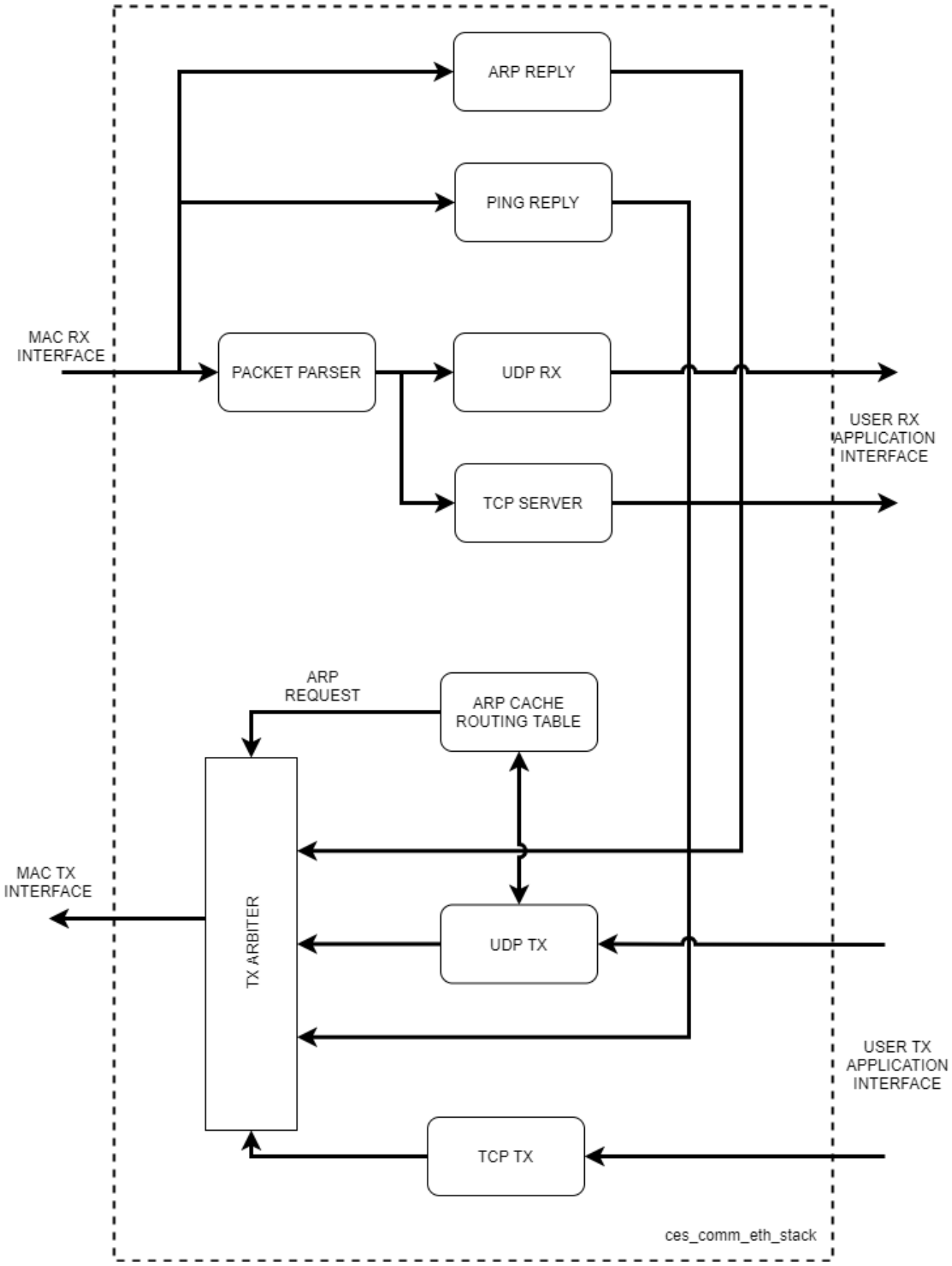

Figure 1 – Ethernet stack block diagram

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.

The code is written in generic VHDL so that it can be ported to a variety of FPGAs capable of running at 125 MHz or above.

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
t_	Type
_st	FSM state

2 PRODUCT CONFIGURATION

The key configuration parameters are brought to the interface so that the user can change them dynamically at run-time.

Pre-synthesis configuration parameters

- Number of TCP Streams
- Number of TX UDP Channel
- Number of RX UDP channel
- Routing table refresh period
- TX/RX elastic buffer size

Run-time configuration parameters

Run-time configuration	*Description*
MAC address MAC_ADDR(47:0)	This network node 48-bit MAC address. For each instantiation, the user is responsible for selecting a unique 'hardware' address.. Natural bit order: enter x0123456789ab for the MAC address 01:23:45:67:89:ab.
IPv4 address IPv4_ADDR(31:0)	Local IP address. 4 bytes for IPv4. Byte order: (MSB)192.168.0.1(LSB)
Subnet Mask SUBNET_MASK(31:0)	Subnet mask Byte order: (MSB)255.255.255.0(LSB)
Gateway IP address GATEWAY_IP(31:0)	One gateway through which packets with a WAN destination are directed. Byte order: (MSB)192.168.1.1(LSB)
TCP port numbers	Identifies the TCP ports used
TX UDP port numbers	Identifies the TX UDP ports used
RX UDP port numbers	Identifies the RX UDP ports used

3 IP-CORE INTERFACE

There are three main signal groups in this interface:

- MAC interface (direct connection to Campera-ES MAC core or equivalent)
- UDP frames or UDP streams to/from the user application.
- TCP streams

All signals are synchronized to a user-defined clock. A minimum clock speed of 125 MHz is required to guarantee a 1 Gbps throughput.

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_clk_frequency	User clock frequency (at least 125 MHz)
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.

The port description above has one TCP stream and 1 UDP TX/RX frame

Table 2 – ports description

Port name	Description
clk_i	User clock
rst_n _i	User reset, synchronous with clk_i, active low
mac_addr_i[47 downto 0]	This network node 48-bit 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 It is essential that this input matches the MAC address used by the MAC/PHY
ipv4_addr_i[31 downto 0]	Local IP address. 4 bytes for IPv4. Byte order: (MSB)192.68.1.30(LSB)
ipv6_addr_i[127 downto 0]	
subnet_mask_i[31 downto 0]	Subnet mask to assess whether an IP address is local (LAN) or remote (WAN) Byte order: (MSB)255.255.255.0(LSB)
gateway_ip_addr_i[31 downto 0]	One gateway through which packets with a WAN destination are directed. Byte order: (MSB)192.68.1.1(LSB)
mac_tx_data_o[7 downto 0]	MAC reads the data at the rising edge of clk_i when mac_tx_data_valid_o = '1'
mac_tx_data_valid_o	TX Data valid
mac_tx_eof_o	End Of Frame

mac_tx_cts_i	Clear To Send, flow control signal, indicating room in the MAC tx elastic buffer for a complete maximum size frame 1518B. The user should check that this signal is high before deciding to send sending the next frame. mac_tx_cts_i may go low while the frame is transfered in. Ignore it as space is guaranteed at the start of frame.
mac_rx_data_i[7 downto 0]	User reads the data at the rising edge of clk_i when mac_rx_data_valid_i = '1'
mac_rx_data_valid_i	RX Data valid
mac_rx_sof_i	Start of Frame, when '1' mark the first byte in a received frame
mac_rx_eof_i	End of Frame, when '1' mark the last byte in a received frame
udp_rx_data_o[7 downto 0]	UDP RX Data, valid when udp_rx_data_valid_o = '1'
udp_rx_data_valid_o	UDP RX valid data
udp_rx_sof_o	UDP RX Start Of Frame
udp_rx_eof_o	UDP RX End Of Frame
udp_rx_dest_port_no_i[15 downto 0]	UDP RX Port Number (1-65535)
udp_tx_data_i[7 downto 0]	UDP TX Data, valid when udp_tx_data_valid_i = '1'
udp_tx_data_valid_i	UDP TX valid data
udp_tx_sof_i	UDP TX Start Of Frame
udp_tx_eof_i	UDP TX End Of Frame
udp_tx_cts_o	Clear To Send, the UDP TX Core is ready to accept a new frame
udp_tx_ack_o	Previous UDP frame has been successfully sent
udp_tx_nak_o	Previous UDP frame error
udp_tx_dest_ip_addr_i[127 downto 0]	Destination IP Address
udp_tx_dest_port_no_i[15 downto 0]	UDP Destination Port [1-65535]
udp_tx_source_port_no_i[15 downto 0]	UDP Source Port [1-65535]
tcp_rx_data_o[7 downto 0]	TCP received data bytes TCP → User Application
tcp_rx_data_valid_o	TCP RX stream data valid
tcp_rx_rts_o	TCP RX stream Request To Send
tcp_rx_cts_i	User supplied Clear to Send
tcp_tx_data_i[7 downt o0]	User Application → TCP TX data bytes
tcp_tx_data_valid_i	TCP TX data valid
tcp_tx_cts_o	TCP TX Clear to Send
tcp_connected_flag_o	TCP stream up indicator

4 GENERAL DESCRIPTION

The root entity contains instantiations of the IP protocols and a transmit arbitration mechanism to select the next packet to send to the MAC/PHY.

The root also includes the following components:

- The PACKET PARSER component parses the received packets from the MAC and efficiently extracts key information relevant for multiple protocols. Parsing is done on the fly without storing data.
- The ARP component detects ARP requests and assembles an ARP response

Table 3 – ARP packet structure

Octet offset	0	1	Value
0	Hardware type (HTYPE)		Ethernet: 0x0001
2	Protocol type (PTYPE)		IP: 0x8000
4	Hardware address length (HLEN)	Protocol address length (PLEN)	Hardware size:6 Protocol Size: 4
6	Operation (OPER)		Request: 0x0001 Reply: 0x0002
8	Sender hardware address (SHA) (first 2 bytes)		
10	(next 2 bytes)		
12	(last 2 bytes)		
14	Sender protocol address (SPA) (first 2 bytes)		
16	(last 2 bytes)		
18	Target hardware address (THA) (first 2 bytes)		
20	(next 2 bytes)		
22	(last 2 bytes)		
24	Target protocol address (TPA) (first 2 bytes)		
26	(last 2 bytes)		

- Ethernet packet for transmission to the MAC
- The PING component detects ICMP echo (ping) requests and assembles a ping echo Ethernet packet for transmission to the MAC. Instantiated once.

Table 4 – PING IPv4 Datagram structure

	Bits 0–7	Bits 8–15	Bits 16–23	Bits 24–31
Header (20 bytes)	Version/IHL	Type of service (ToS)	Length	
	Identification		*flags* and *offset*	
	Time to live (TTL)	Protocol	Header checksum	

	Source IP address		
	Destination IP address		
ICMP header (8 bytes)	Type of message	Code	Checksum
	Header data		
ICMP payload (*optional*)	Payload data		

The PING request is an ICMPO message with Type = 8 (IPv4) and Code = 0, whereas the PING reply has a Type = 0 (IPv4) and Code = 0.

- The adaptable UDP TX component encapsulates a data packet into a UDP frame that may be addressed to any port or IP address. Regardless of the number of source or destination UDP ports, only one instance is created.

Table 5 – UDP Datagram header

Offsets	Octet	0	1	2	3
0	0	Source port		Destination port	
4	32	Length		Checksum	

- The UDP RX. component verifies and extracts data packets from received UDP frames. The validity confirmation is made accessible at the end of the packet since the validation is done on the fly (no storage) when incoming data is travelling through. When an invalid packet is received, the caller program should be allowed to 'backtrack.' Regardless of the number of UDP ports being listened to, only one instance is created.
  Although this component was designed for a single port, it may simply be updated to support multiple ports (follow the PORT NO signal). As a result, there's never a need to instantiate several components.

The TCP protocol's heart is the TCP SERVER component. It's written in a parametric way to handle NTCPSTREAMS simultaneous TCP connections. It basically acts as a TCP server's state machine, listening for connection requests from remote TCP clients, establishing and breaking down connections, and handling flow control while connections are established.

- TCP TX is a component that formats TCP tx frames at all layers: TCP, IP, and MAC/Ethernet. It affects all concurrent streams in the same way.
  TCP tx payload data is stored in distinct elastic buffers in the TCP TXBUF component, one for each transmit stream. Prior to synthesis, the buffer size is customizable as NBUFS*16Kbits RAM blocks.
- The component TCP RXBUFNDEMUX demultiplexes multiple TCP rx streams. It is devoid of any elastic buffer. If the program requires it, it is assumed that the user will create elastic buffers. Without gaps or backtracking, data bytes are received in order.
- WHOIS: Before sending any IP packet, the destination IP address must be converted to a 48-bit MAC address.

  For this reason, a look-up table (inside arp cache module) is available. When the look-up table does not contain an entry for the destination IP address, an ARP request is broadcast to all, requesting that the receiver react with an ARP response. The whois2.vhd component's principal job is to build and transmit this ARP request.

- ARP CACHE: To store 128 MAC/IP/Timestamp records, a block RAM is employed as cache memory. Each record has
  - a 48-bit MAC address,
  - a 32-bit IPv4 address, and
  - a timestamp for when the data was last updated.

  The data is updated on a regular basis depending on received ARP answers and IP packets. The component keeps track of the oldest record, which will be overwritten next.

  This component searches the block RAM for a matching IP address key whenever the application asks the MAC address for a specific IP address (search key). If a match is detected, the related MAC address is returned. This component asks WHOIS module to send an ARP request packet if the search key is not found or is older than a refresh interval.

  The code has been optimized for speed. The response time varies between 0.1 and 1.3 micro seconds, depending on the position of the record in memory.

  This routing table is created once and used by various instances that need routing services.

An arbitration circuit is used to order routing requests from multiple transmit instances (for example, multiple UDP TX component instantiations).

4.1 FUNCTIONAL TIMING

The input interface is a simple framing interface with start of frame, data valid and end of frame for the UDP or TCP data.

A clear to send signal is used for the handshake with the MAC module.

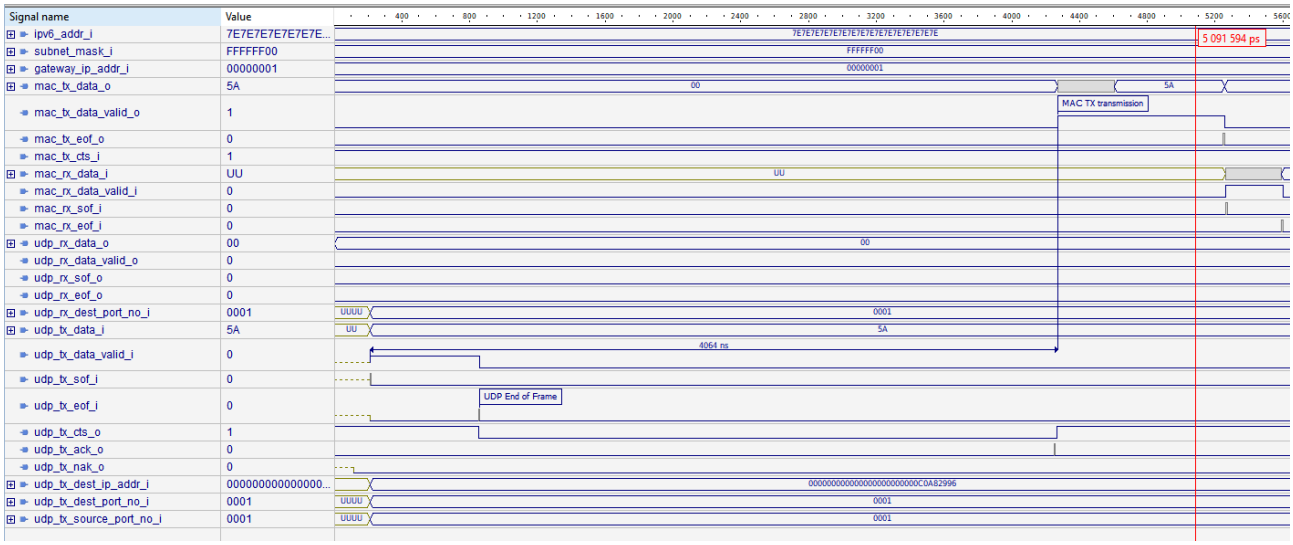

Figure 2 – UDP TX frame and ethernet frame to the MAC

5 SIMULATION

Testbenches are available for the following protocols:

- ARP
- PING
- UDP
- TCP

6 SYNTHESIS

The following tables show some synthesis tests. Here examples are given of the resource utilization in selected standard FPGA's.

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

Family Device	LCs	Registers	Memory	Mult/DSP
Intel Cyclone 10	3390 (ALMs)	3393	192512 (Memory Bits)	0
Xilinx Kintex US	2935 (LUTs)	3341	6 BRAM	0

Table 6. 1 UDP RX/TX frame channel and 1 TCP stream

7 10 GbE

The 10 GbE version of the core needs a minimum clock of 156.25 MHz and has the same structure and features of the 1 GbE version. The main architectural difference is that 8 samples per clock cycles must be processed

Family Device	LCs	Registers	Memory	Mult/DSP
Xilinx Kintex US	3173 (LUTs)	5916	32.5 BRAM	0

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

REVISIONS			
REV.	DESCRIPTION	MODIFIED BY	DATE
1.0	Initial release		
1.1	New template porting		