UDP/IP Transmitter/Receiver Lite
- High Speed Communication Core

October, 2007 Product Specification

Prevas AB
Box 4, Visit: Klockartorpsgatan 14
S – 721 03 Vasteras
Sweden

Phone: +46 (0)21 – 360 19 00
Fax: +46 (0)21 – 360 19 29
Email: stefan.sjoholm@prevas.se
URL: www.prevas.se

Features

- Send and receive UDP-packets (User Datagram Protocol) over Ethernet
- Send and receive broadcast messages
- Answer to echo ping-requests
- Check incoming messages against CRC errors
- Send 256 byte wide packets
- Supports transfer rates at 10/100 Mbit/s
- Fully synchronous implementation, which uses block-RAM for storage
- Compatible with standard Ethernet transceivers MII (Media Independent Interface)

Applications

The core is suitable for implementing high-speed point-to-point communication with a PC where a UART or the parallel port is not fast enough. It is also great for sending large amount of data (210 bytes) in one packet.

General Description

The UDP/IP core sends and receives Ethernet packets to a 10/100Mbit transceiver. The core has a unique MAC-address (Media Access Control) and a specified IP-address (Internet Protocol).

It will respond to ARP requests that is targeted to the core's IP-address and sent as Ethernet broadcast. The purpose of this packet is for other Ethernet nodes to obtain the core's MAC-address.

The IP core will send an ARP-request (Address Resolution Protocol) to target the IPC-address after reset.

Echo request messages from other PC:s running in an windows environment, will be answered by a echo reply message. When UDP packets are sent to the core's IP-address, it will be received and stored in RAM. Status flags will then be activated so that the system knows when a new packet has arrived.

New UDP-packets is written to the UDP/IP core's RAM memory, when finished the send status flag will be activated and the packet will be transmitted to the PC.

A received packet with faulty CRC or a packet size larger than 256 bytes will not be processed.

CORE Facts

Provided with Core	
Documentation	Core documentation
Design File Formats	EDIF netlist; VHDL Source RTL (available at extra cost)
Constraints Files	Udp_ip.ucf
Verification	VHDL test bench
Instantiation templates	VHDL
Reference designs & application notes	UDP-packet loopback program, with PC-interface
Additional Items	None
Simulation Tool Used	
Modelsim v6.0	
Support	
Support provided by [Prevas AB]	

Functional Description

As shown above and explained below, the UDP/IP core includes six major blocks: Receiver, Transmitter, CRC checker, CRC generator, packet interpreter and packet composer. Furthermore, the design includes memory blocks for message storage.

After the core has been reset it will send an ARP request automatically to the specific PC that the core shall communicate with. The PC will then respond with an ARP reply message.

Family	Example Device	Fmax (MHz)	Slices	IOB	GCLK	Design Tools
Spartan-IIE	XC2S300E-6	56	700	71	2	ISE 6.3i

Table 1: Example Implementation Statistics

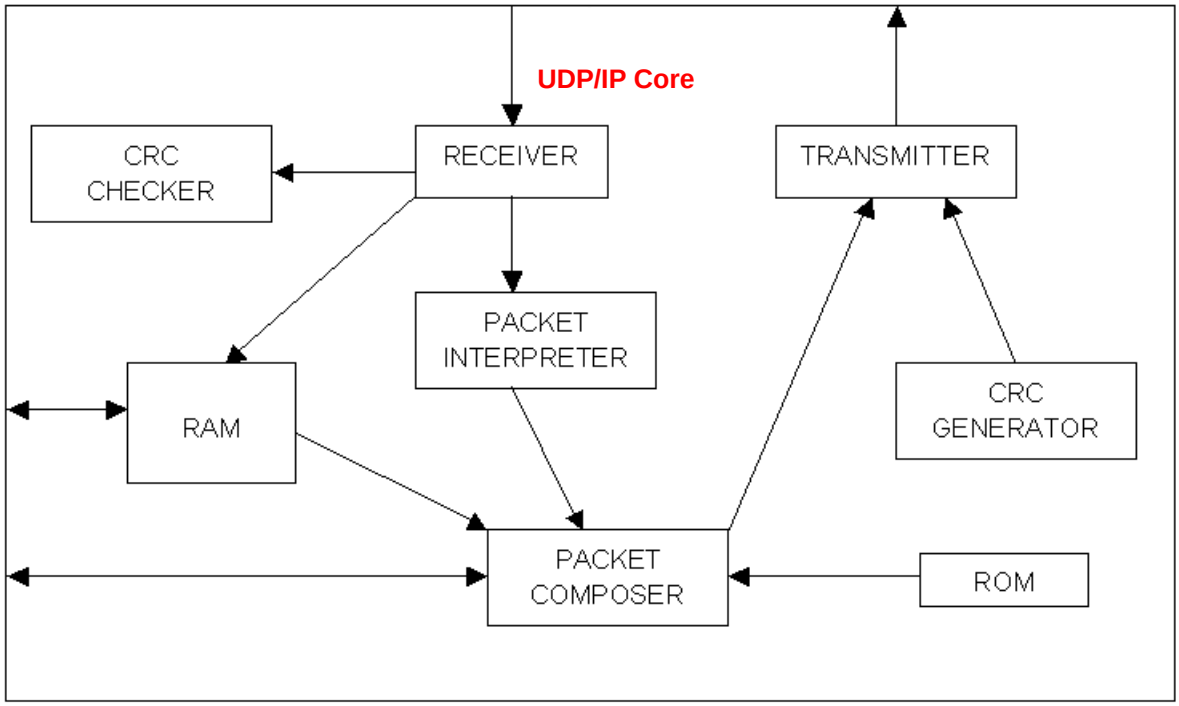

Figure 1: UDP/IP Core Block Diagram

Receiver

This block handles the incoming messages. Data arrives in nibble-packets. It gradually stores the message in RAM. A Message that is larger than 256 bytes will be rejected.

CRC Checker

32 bit CRC is calculated on incoming packets. If it doesn't match with the received CRC, the packet is thrown away and the CRC error flag is set.

Packet Interpreter

This block checks certain fields in the received packet and decides if it is a supported packet or not. If it is a supported packet the block initiate the sending of a response packet. If it was an UDP-packet it will flag to the application layer that a new UDP-packet has arrived.

Packet Composer

In this component the packets are put together. Four types of packets can be sent:

- ARP-requests
- ARP-responses
- Echo-responses
- UDPs

Depending one which packet that will be sent, different actions are taken. The majority of each message is read from a ROM and copied to RAM. After that, other message components like checksums, lengths, identifier etc are added to the message. UDP messages will be put together after the UDP transmit request signal has been set from the application layer.

Transmitter

This block reads data from RAM and puts out the transmit packet to the nibble-wide databus and sets control signals. At the beginning, 16 nibbles of preamble are sent, where the last one is a start of frame nibble. At the end of each packet the 32 bits CRC checksum is sent.

CRC Generator

While the packet is sent, CRC is calculated, one byte at a time. It uses the CRC32 polynomial for Ethernet. The polynomial looks like this:

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 + 1.

RAM/ROM

The base of all messages is stored in block ROM. There are four separate ROM:s, one for each message.

The memory is 512*8 bits wide. It stores the incoming message, holds the transmit message temporarily and stores UDP-data from the superior application.

Core Modifications

The size of the UDP-packet can be extended. Echo request/response functionality can be excluded. Other types of packets can be supported. CRC32 checking on incoming packets can be omitted.

UDP/IP I/O Signals

The signal names of the UDP core are shown in Table 2

Signal	Direction	Description
reset_n	Input	Master Reset (Asynchronous)
phy_reset_n	Output	Transceiver master reset
rxdata(3:0)	Input	Transceiver receiver databus
rx_clk	Input	Receiver master clock from transceiver
rx_dv	Input	Receiver data valid strobe from transceiver
tx_clk	Input	Transmitter master clock from transceiver
txdata(3:0)	Input	Transceiver transmitter databus
tx_en	Output	Transmit data enable to transceiver
tx_er	Output	Transmit error line
crc_error	Output	Activated when CRC error received
phy_res	Input	Input for transceiver reset
led	Output	Toggles for every new UDP-packet rec.
ram_clk	Output	Ram memory clk
udp_pkt_rec	Output	New udp packet received
udp_pkt_aq	Input	Packet received acknowledge
udp_data_rec(7:0)	Output	Databus for reading packet from memory
udp_data_tra(7:0)	Input	Databus for writing UDP-packet to RAM
udp_addr(8:0)	Input	Addressbus for UDP-RAM memory
ram_wr_str	Input	RAM write strobe
udp_tra_req	Input	UDP transmit request
udp_tra_acq	Output	UDP transmit acknowledge
udp_length(7:0)	Input	UDP transmit packet length

Table 2: UDP/IP I/O Signals

Core Assumptions

The UDP/IP core follows the Ethernet and IP standard but has the following simplifications:

- The PC is assumed to have a specific IP-address, but any MAC-address is suitable.
- Packets bigger than 256 bytes will not be received.
- The core will not work for 1Gbit Ethernet.
- Min packet length is 64 bytes; therefore shorter packets will be padded with extra bytes of any value from RAM at the end.
- Echo and UDP messages will not be checked for correct UDP/ICMP checksum.
- The IP header checksum will not be checked.
- On Echo-requests, IP header identifier field will not be increased.

Verification Methods

The UDP/IP core's functionality has been extensively tested with a testbench and a large number of test patterns. It has also been implemented on a Spartan-IIE with a Broadcom transceiver, where the communication with a PC has been verified.

Design Services

Prevas also offers core integration, core ustomisation and other design services.

Ordering Information

This product is available from Prevas AB, under erms of the SignOnce IP License. See www.prevas.se for pricing or contact Prevas for additional information about this product.

Prevas AB
Box 4, Visiti: Klockartorpsgatan 14
S – 721 03 Vasteras
Sweden

Email: stefan.sjoholm@prevas.se
Phone: +46 (0)21 – 360 19 00
Fax: +46 (0)21 – 360 19 29
URL: www.prevas.se

Prevas AB cores are purchased under a Licence Agreement, copies of which are available on request. Prevas AB retains the right to make changes to these specifications at any time, without notice. All trademarks, registered trademarks, or servicemarks are the property of their respective owners.

Related Information

Xilinx Programmable Logic

For information on Xilinx programmable logic or development system software, contact your local Xilinx sales office, or:

Xilinx, Inc. 2100 Logic Drive
San Jose, CA 95124
Phone:+1 408-559-7778
URL: www.xilinx.com