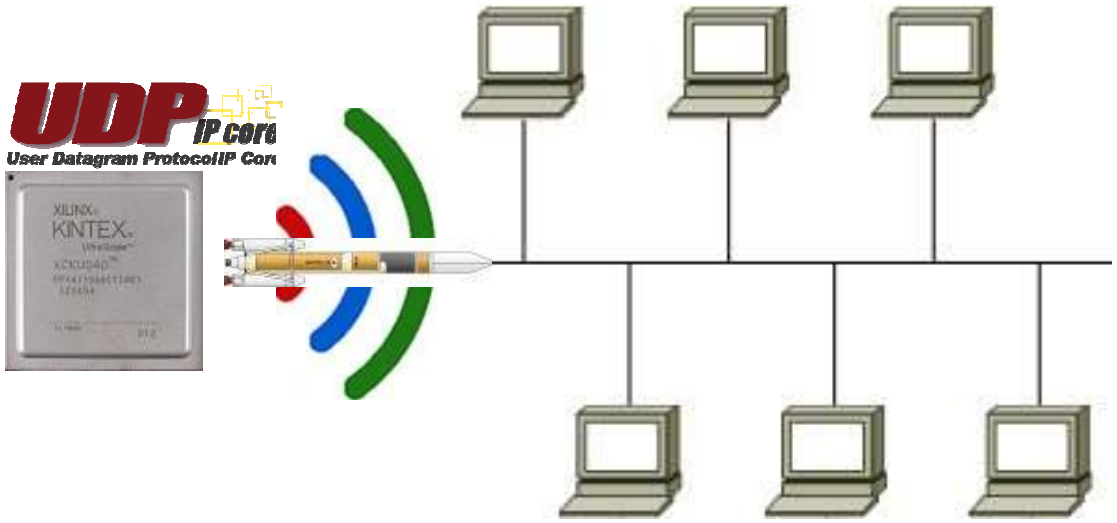
UDP10G-IP Introduction (Xilinx)

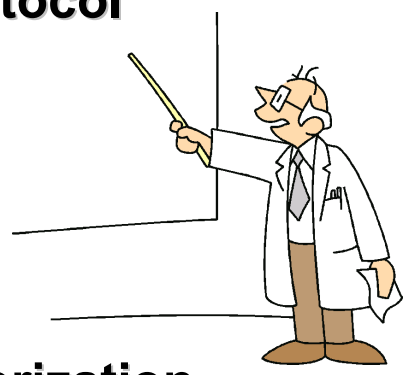
Ver1.1E



Super UDP Speed by hard-wired IP-Core

Agenda

- Merit and demerit of UDP protocol
- UDP10G-IP core overview
- UDP10G-IP core description
  - Initialization
  - High-speed transmit
  - High-speed reception
- User I/F, Buffer size parameterization
- Reference design
- Resource usage and real performance
- Application example

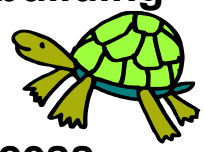


Merit and demerit of UDP protocol

- **Merit**
  - High-speed and low-latency by minimum overhead
  - Supports 1-to-N multicast and 1-to-All broadcast
  - Suitable for real-time application such as VOD system
- **Demerit**
  - No ACK/retransmit, so data reliability is not guaranteed
  - If reliability is necessary, application layer must support it

UDP implementation problem by CPU

- **Problem in performance and latency**
  - CPU resource consumption by UDP packet building
    - Check-sum calculation
    - Concatenate header and transmit data
  - Bandwidth is not stable due to firmware process

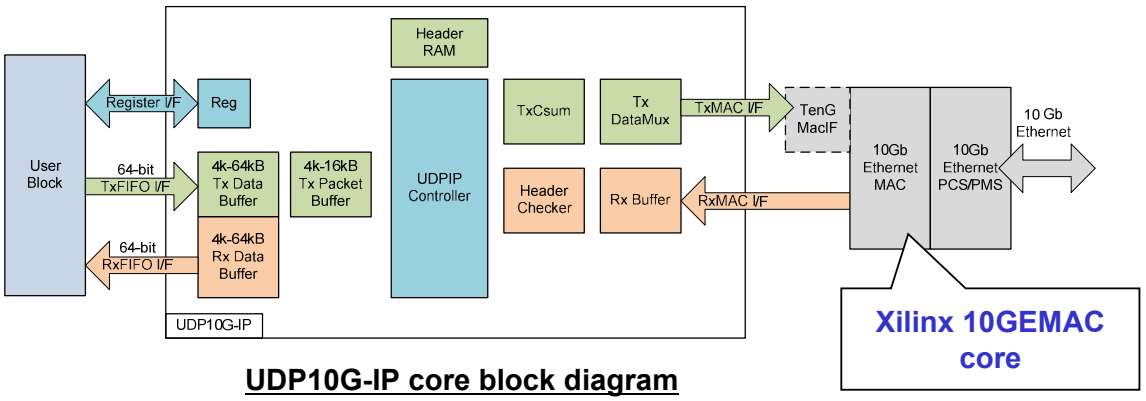


- **The problem gets even worse with full duplex**
  - CPU needs to process time sharing between Tx&Rx
  - Bandwidth and latency further drops
  - Fatal problem for real time application

UDP10G-IP core can provide ideal solution!

UDP10G-IP core Overview

- Fully hard-wired UDP control for both Tx and Rx
- Inserts between user logic and Xilinx 10GEMAC module
- Supports Full Duplex communication



UDP10G-IP core block diagram

UDP10G-IP core Advantage 1

- Fully hard-wired UDP protocol control
  - Possible to build CPU-less network system
  - Zero load for CPU

- Support all of Tx only, Rx only, and full-duplex
  - More than 1200MByte/sec real performance

- Can even keep some data reliability
  - Tx: Calculate check sum and build header automatically
  - Rx: Discard received Packet if check sum does not match

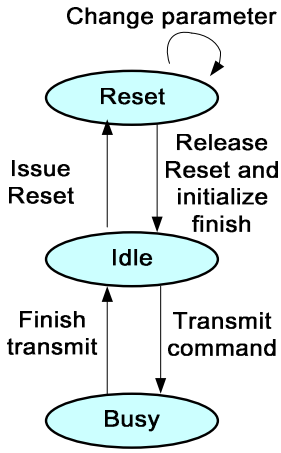
UDP10G-IP core Advantage 2

- **Selectable data buffer size**
  - Selectable buffer size of memory usage vs. performance
- **Supports IP fragment packet reception**
  - Receive IP fragment packet when packet order is correct
- **Reference design on Xilinx evaluation board**
  - Full Vivado project for standard Xilinx board
  - Free bit-file for evaluation before purchase
  - All source code (except IP-core) in design project

- **Can support multicast/broadcast transmission**
  - Provided by IP-core customization service

UDP10G-IP core Operation

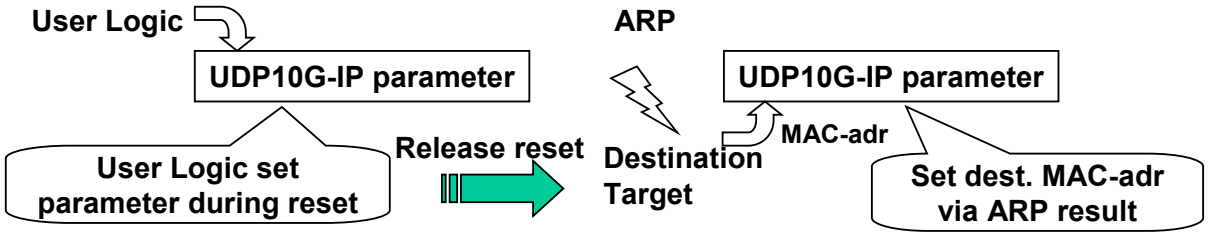
- Set parameter (IP-adr&MAC-adr, etc) during Reset
- Release Reset then initialize including ARP
- Idle state after initialization finish, wait command
- Tx operation starts by user command
- Rx operates at any time except Reset state
  (Accepts all Rx packet if parameter match)
- Tx and Rx operates individually (full duplex)
- If want change parameter, move to Reset state
  (transfer/packet length can change except Busy)



State Diagram

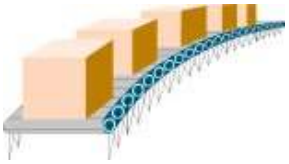
UDP10G-IP Initialization

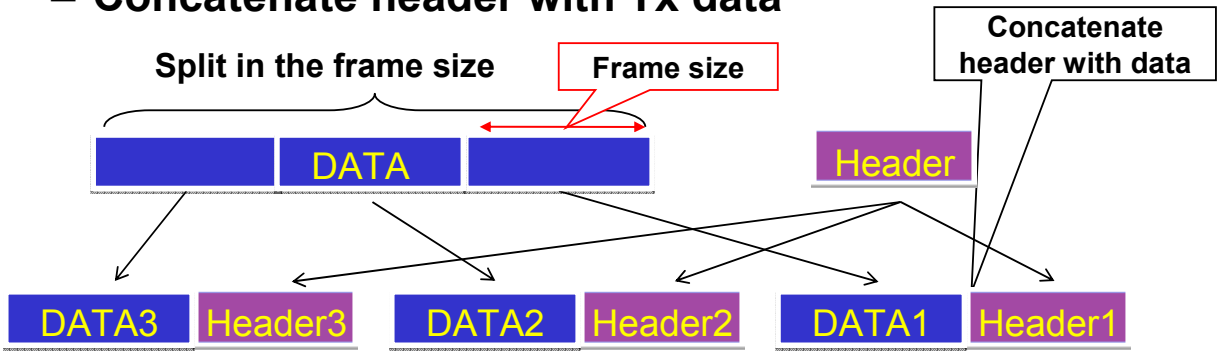
- Set parameter to UDP10G-IP
  - User logic can set parameter during UDP10G-IP reset
  - Set IP address, MAC address, and Port number
  - Release reset after parameter setting finish
- UDP10G-IP executes ARP after reset release
  - Issue ARP to destination target when Client mode
  - Wait ARP reception when Server mode



High-Speed Tx

- Tx Packet Generation
  - User Logic writes Tx data to TxFIFO
  - Split Tx data in the frame size
  - Calculate check sum and set to the header
  - Concatenate header with Tx data

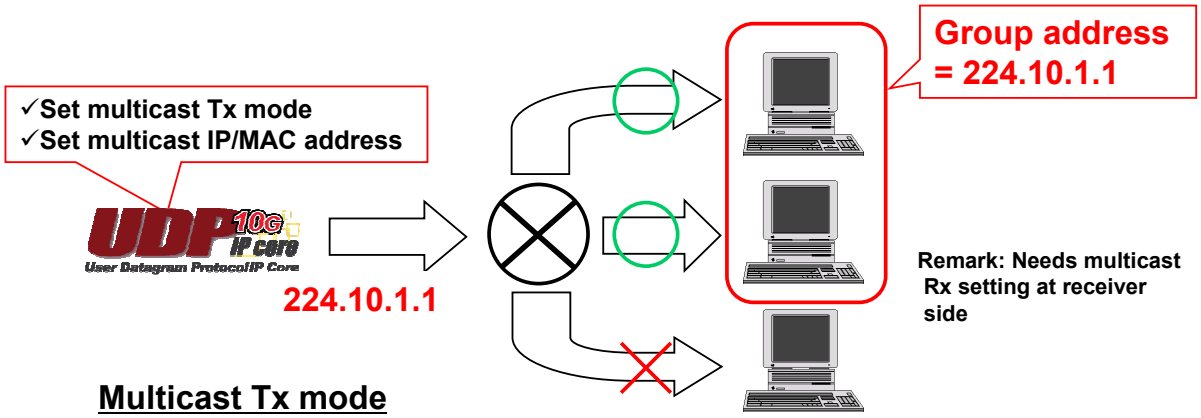




Tx Packet Generation

Multicast/Broadcast High-Speed Tx (optional)

- Multicast/broadcast Tx via customization
  - Suppress automatic ARP execution
  - Set multicast IP/MAC address from user logic

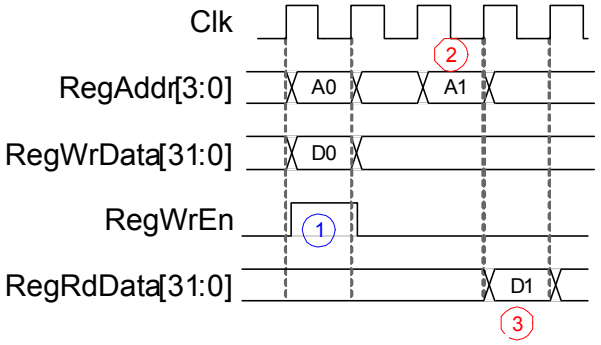


High-Speed Rx

- Rx packet header check
  - Verify all of MAC, IP, and UDP header
  - Receive IP fragment packet when order is correct
- Check sum calculation and verification
  - Calculate check sum in received packet
  - Verify calculated value with header value
  - When mismatch, packet data is discarded

User Interface (Control)

- 3 types of Register I/F, Tx FIFO I/F, and Rx FIFO I/F
  - Register I/F for initial parameter setting and Tx/Rx command
  - Tx FIFO I/F and Rx FIFO I/F is standard FIFO interface

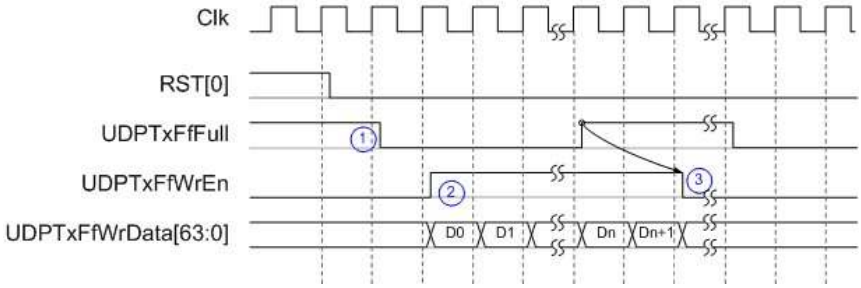


Register I/F timing

[Register Write]
(1) Assert RegWrEn with RegAddr and RegWrData

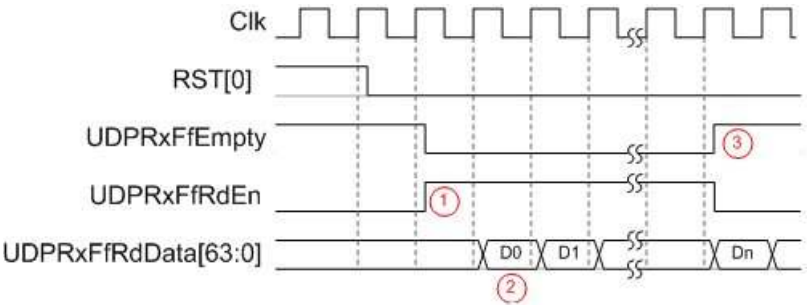
[Register Read]
(2) Set RegAddr
(3) Valid RegRdData output in the next clock

User Interface (Data)



Tx FIFO I/F timing

[Tx data write]
(1) Check FIFO is not full
(2) Write data with WrEn
(3) Suspend write within 4 Clk after Full assertion



Rx FIFO I/F timing

[Rx data read]
(1) Read by RdEn assertion when not Empty
(2) Read data after 1 Clk
(3) Read is inhibited when Empty

Buffer Capacity

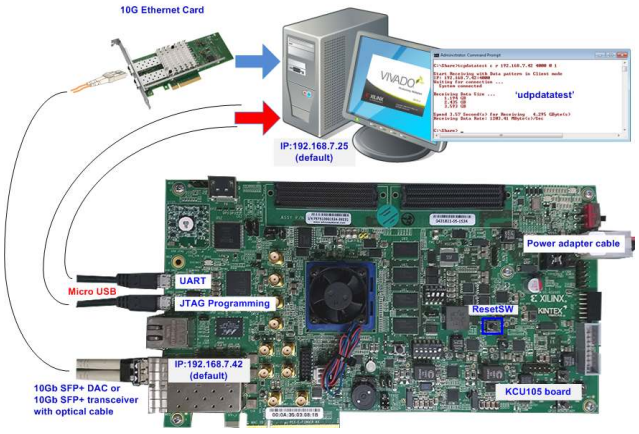
- **Parameterized 3 types of data buffer**
  - **(1) Tx Data Buffer: 4KByte - 64KByte**
  - **(2) Tx Packet Buffer: 4KByte - 16KByte**
  - **(3) Rx Data Buffer: 4KByte - 64KByte**
- **User can optimize resource usage and performance**

Generic Name	Range	Description
TxBufBitWidth	9-13	Set Tx data buffer size in address bit width When set to 9, size is 4KByte, when 13, 64KByte for example.
TxPacBitWidth	9-11	Set Tx packet buffer size in address bit width When set to 9, size is 4KByte, when 11, 16KByte for example
RxBufBitWidth	9-13	Set Rx data buffer size in address bit width When set to 9, size is 4KByte, when 13, 64KByte for example.

Buffer size is selectable by parameterization

Free Bit File for Evaluation

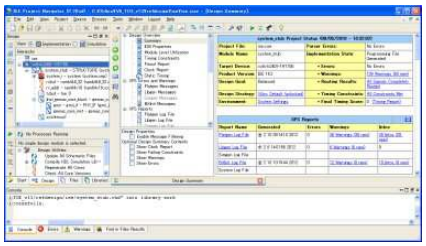
- **Bit file for evaluation with Xilinx standard board**
  - **Real communication check between FPGA board and PC**
  - **Measure transfer speed performance and data reliability**



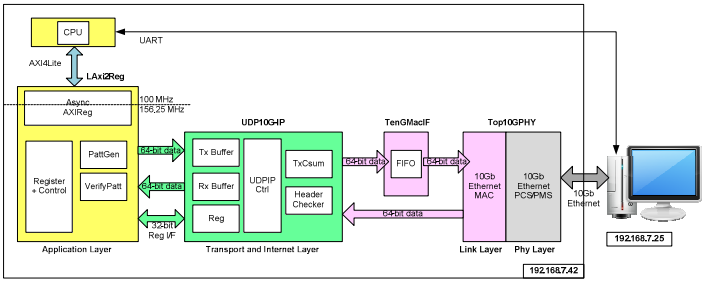
Evaluation environment for Xilinx board (KCU105)

Reference Design Overview

- **Vivado design project for real operation**
  - **Implemented into standard Xilinx board for each device family**
  - **IP-core deliverables include design of evaluation bit file**
  - **All source code (except IP-core) included in full project**



Vivado/EDK project in package



Reference design block diagram

Effective Development on Ref. Design

- **Vivado project is attached to UDP10G-IP package**
- **Full source code (VHDL) except IP core**
- **Can save user system development duration**
  - **Confirm real board operation by original reference design.**
  - **Then modify a little to approach final user product.**
  - **Check real operation in each modification step.**

Short-term development is possible without big turn back

Resource Usage

- **UDP10G-IP core standalone resource usage**
  - **Condition = Maximum buffer setting**

    (TxDataBuf=RxDataBuf=64KB, TxPacketBuf=16KB)

Example Implementation Statistics (7 family)

Family	Example Device	Fmax (MHz)	Slice Regs	Slice LUTs	Slices[1]	BRAMTile[2]
Kintex-7	XC7K325TFFG900-2	156.25	1874	2098	772	36
Virtex-7	XC7VX485TFFG1761-2	156.25	1874	2093	760	36
Zynq-7000	XC7Z045FFG900-2	156.25	1874	2099	784	36

Example Implementation Statistics (Ultrascale family)

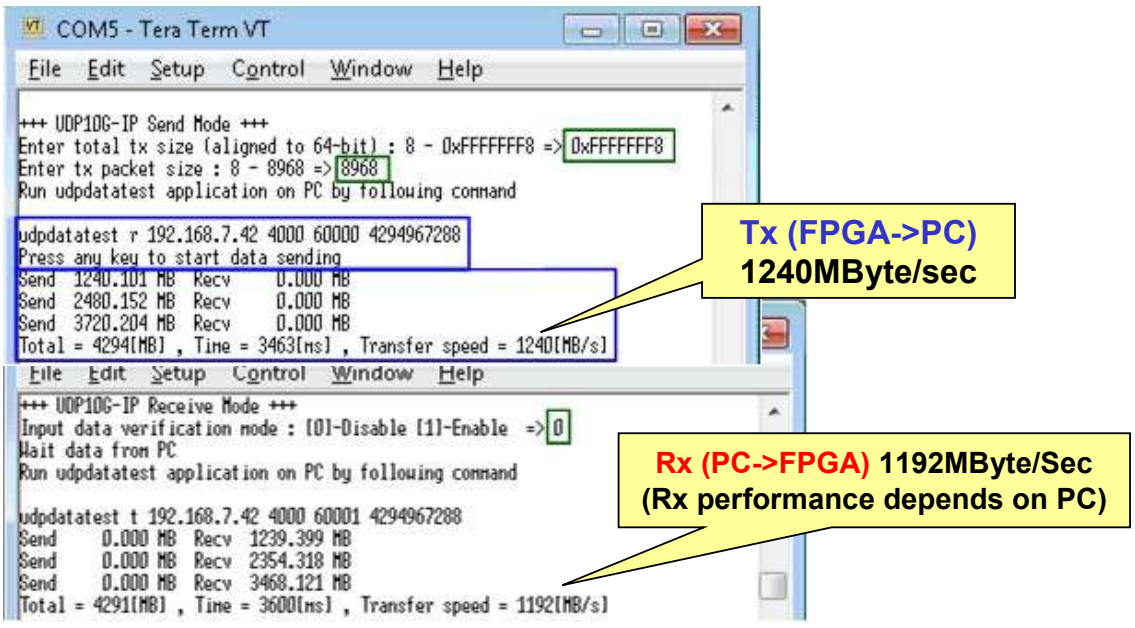
Family	Example Device	Fmax (MHz)	Slice Regs	Slice LUTs	CLB[1]	BRAMTile[2]
Kintex-UltraScale	XCKU040FFVA1156-2E	156.25	1862	2147	455	34.5

UDP10G-IP core standalone compilation result

This result is based on maximum buffer size setting.
User can save memory resource by smaller buffer size setting

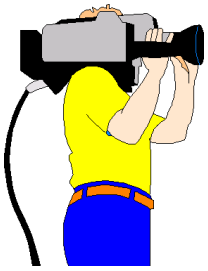
Transfer Performance

- **Real performance in data Tx and data Rx**



UDP10G-IP Application

- **Video-on-Demand via Broadcast**
  - Stream video transmission in real time
  - Requires minimum overhead and latency
  - UDP10G-IP provides best solution



- **Real time Online game**
  - Full duplex of game data download and user operation data upload
  - Very low latency required for realistic game
  - UDP10G-IP can cover full duplex within minimum latency

For more detail

- **Detailed documents available on the web site**
  - http://www.dgway.com/UDP10G-IP_X_E.html
- **Contact**
  - Design Gateway Co,. Ltd.
  - E-mail : ip-sales@design-gateway.com
  - FAX : +66-2-261-2290

Revision History

Rev.	Date	Description
1.0E	7-Dec-2017	English version initial release
1.1E	26-Mar-2019	Added multicast/broadcast Tx feature customization