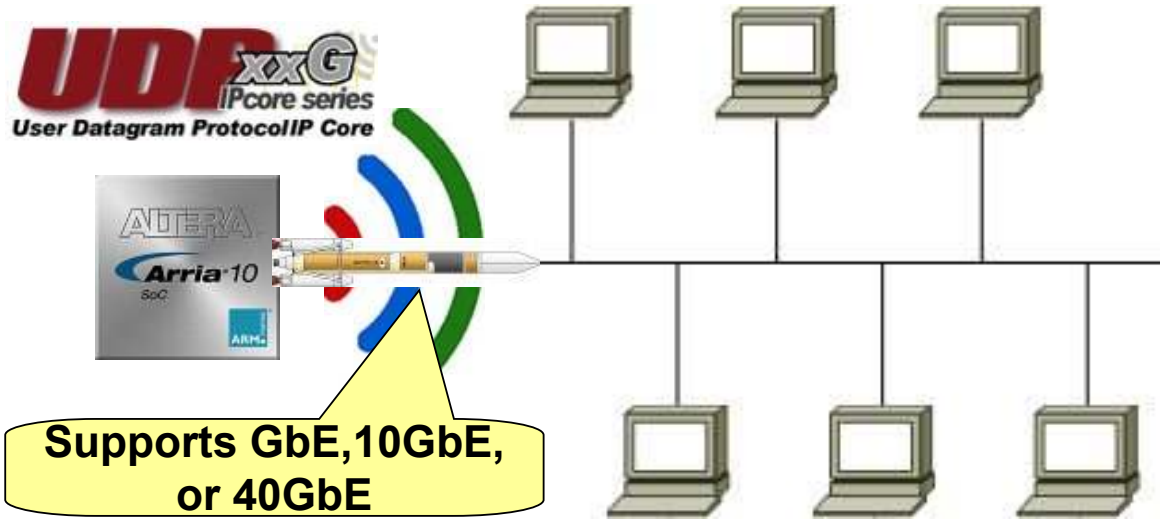
UDPxxG-IP Introduction (Intel)

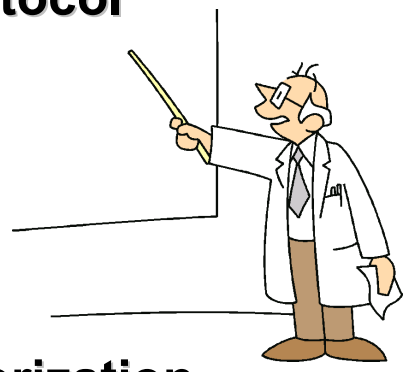
Ver1.0E



Super UDP Speed by hard-wired IP-Core

Agenda

- Merit and demerit of UDP protocol
- UDPxxG-IP core overview
- UDPxxG-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

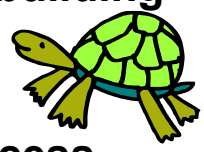


Merit and demerit of UDP protocol

- **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

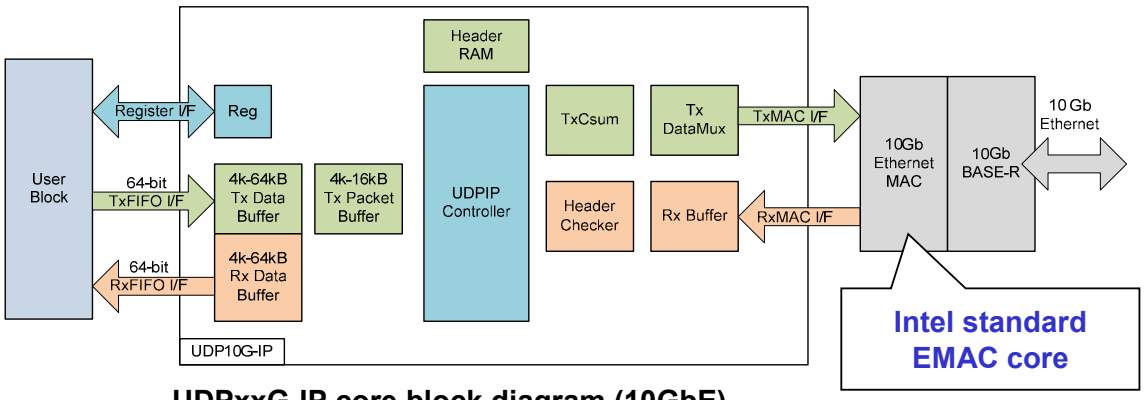


- **The problem gets even worse with full duplex**
  - CPU needs to process time sharing between Tx&Rx
  - Bandwidth and latency further drops
  - Fatal problem for real time application

→ **UDPxxG-IP core can provide ideal solution!**

UDPxxG-IP core Overview

- Fully hard-wired UDP control for both Tx and Rx
- Supports each line rate of GbE, 10GbE, or 40GbE speed
- Inserts between user logic and Intel EMAC module
- Supports Full Duplex communication



UDPxxG-IP core block diagram (10GbE)

UDPxxG-IP core lineup

Family	GbE	10GbE	40GbE
Cyclone V	UDP1G-IP-C5		
Arria V	UDP1G-IP-A5		
Cyclone 10	Possible (Ask)	UDP10G-IP-C10	
Arria 10	UDP1G-IP-A10	UDP10G-IP-A10	UDP40G-IP-A10
Stratix 10	Possible (Ask)	Possible (Ask)	Possible (Ask)

UDPxxG-IP core lineup (as of 1st-Jan-2020)

UDPxxG-IP core Advantage 1

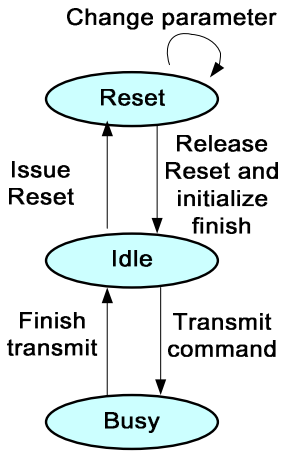
- **Fully hard-wired UDP protocol control**
  - **Possible to build CPU-less network system**
  - **Zero load for CPU**
- **Support all of Tx only, Rx only, and full-duplex**
  - **Actual performance over 90% of line rate**
- **Can even keep some data reliability**
  - **Tx: Calculate check sum and build header automatically**
  - **Rx: Discard received Packet if check sum does not match**

UDPxxG-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 Intel evaluation board**
  - **Full Quartus project for standard Intel board**
  - **Free sof-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**

UDPxxG-IP core Operation

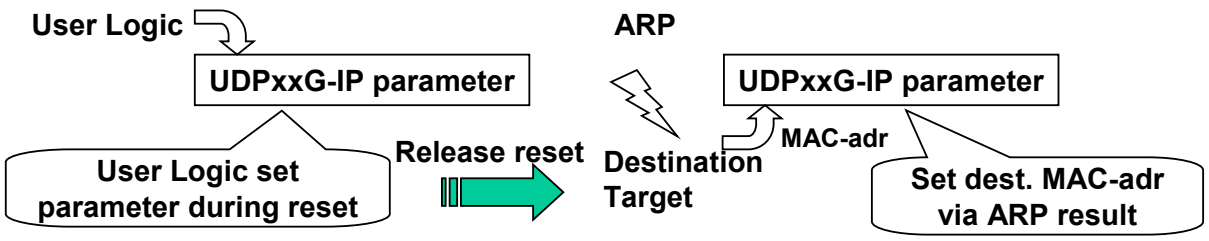
- Set parameter (IP-adr&MAC-adr, etc) during Reset
- Release Reset then initialize including ARP
- Idle state after initialization finish, wait command
- Tx operation starts by user command
- Rx operates at any time except Reset state
  (Accepts all Rx packet if parameter match)
- Tx and Rx operates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



State Diagram

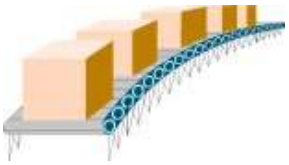
UDPxxG-IP Initialization

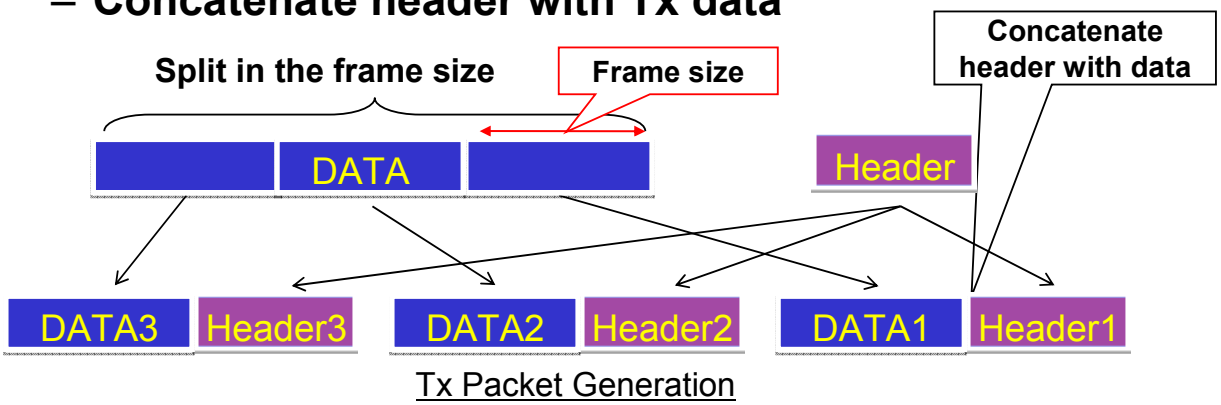
- Set parameter to UDPxxG-IP
  - User logic can set parameter during UDPxxG-IP reset
  - Set IP address, MAC address, and Port number
  - Release reset after parameter setting finish
- UDPxxG-IP executes ARP after reset release
  - Issue ARP to destination target when Client mode
  - Wait ARP reception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

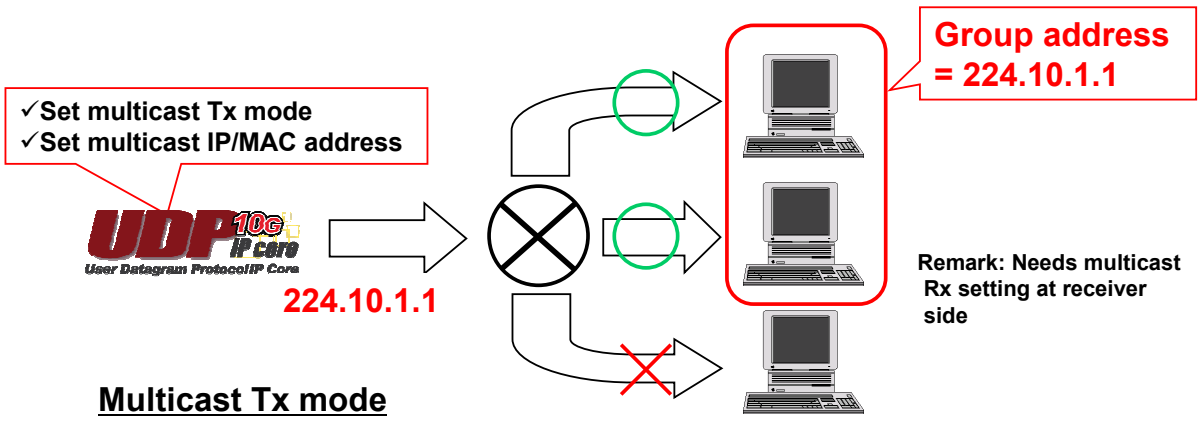




Tx Packet Generation

Multicast/Broadcast High-Speed Tx (optional)

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



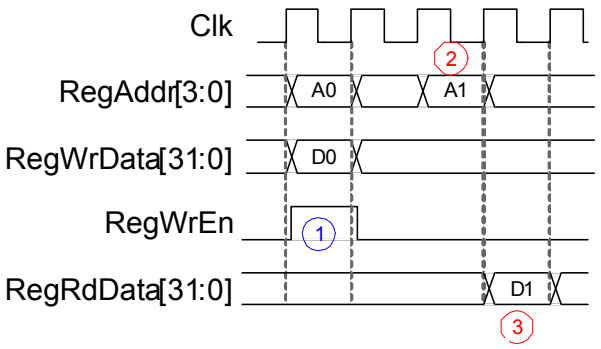
Multicast Tx mode

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

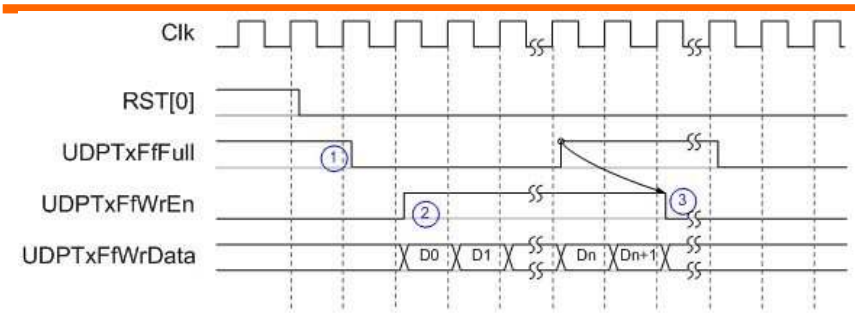


Register I/F timing

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

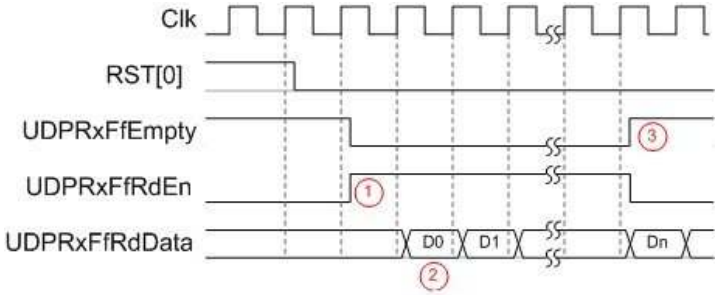
[Register Read]
(2) Set RegAddr
(3) Valid RegRdData output in the next clock

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

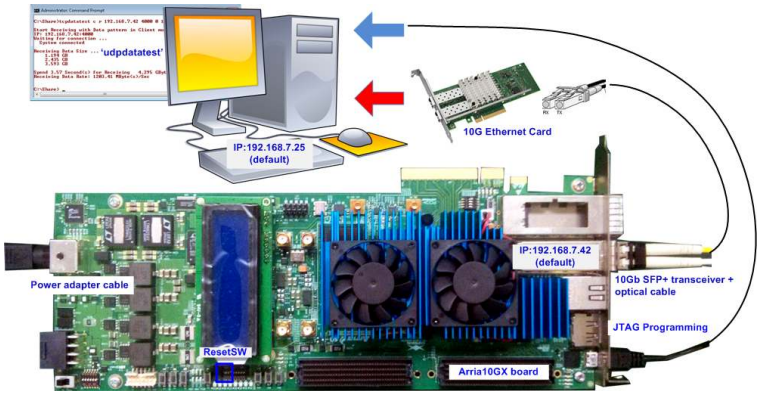
[Rx data read]
(1) Read by RdEn assertion when not Empty
(2) Read data after 1 Clk
(3) Read is inhibited when Empty

Buffer Capacity

- **Parameterized 3 types of data buffer**
  - **(1) Tx Data Buffer: Affects Tx performance**
  - **(2) Tx Packet Buffer: Must set more than max packet size**
  - **(3) Rx Data Buffer: Affects Rx performance**
- **User can optimize resource usage and performance**
  - **Can improve performance if increase buffer size**
  - **Can save FPGA memory resource if reduce buffer size**
  - **Performance and memory usage is trade-off relationship**

Free SOF File for Evaluation

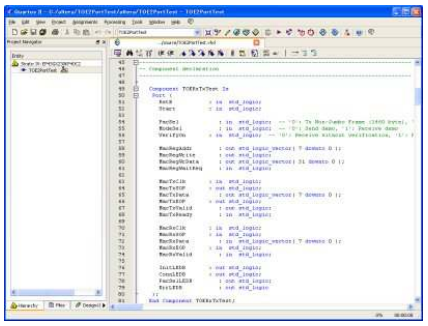
- **SOF file for evaluation with Intel standard board**
  - **Real communication check between FPGA board and PC (Two FPGA boards cross connection for 40GbE case)**
  - **Measure transfer speed performance and data reliability**



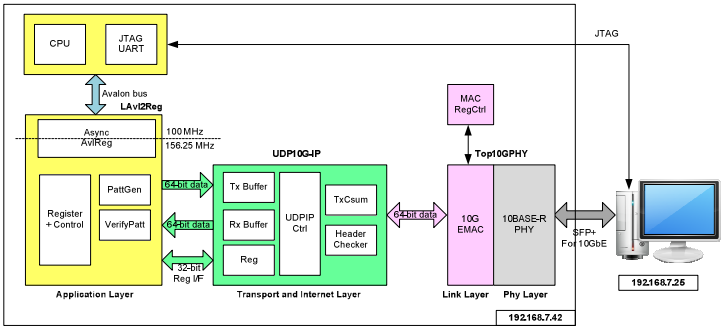
Evaluation environment example using Intel board (Arria10GX)

Reference Design Overview

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



Quartus/Qsys project in package



Reference design block diagram

Effective Development on Ref. Design

- Quartus project is attached to UDPxxG-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

- Each resource usage of GbE/10GbE/40GbE

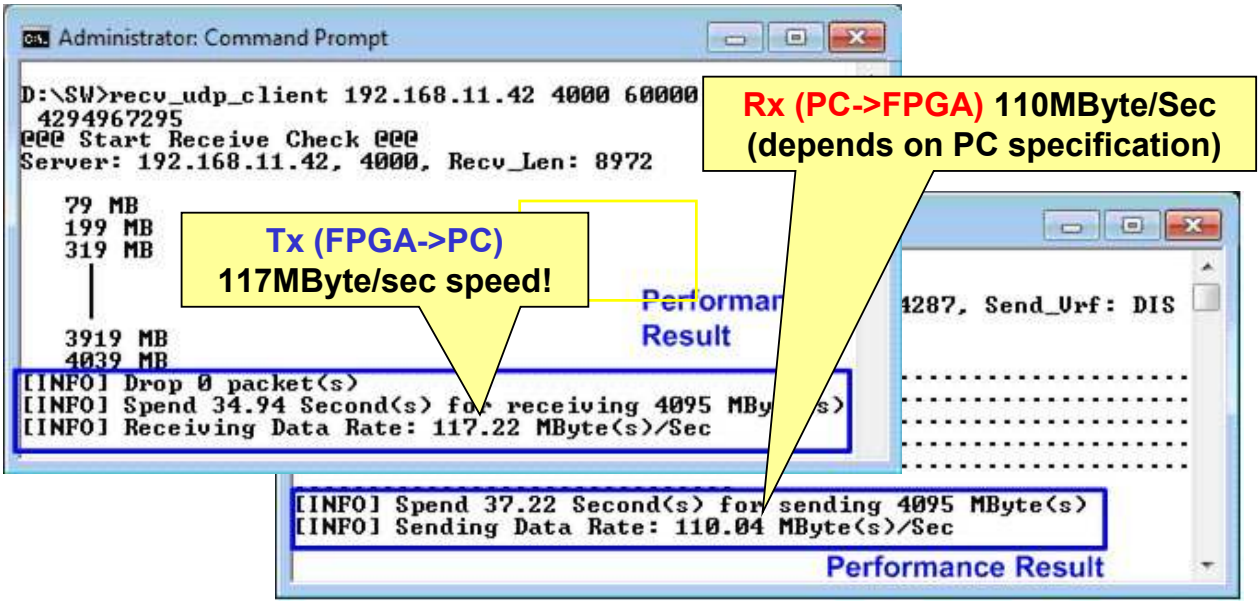
Line Rate (Device)	Clock freq.	Logic resource	Max. memory
GbE (Arria 10)	125MHz	1,051 ALM	1,181,696 bit
10GbE (Cyclone 10)	156.25MHz	1,348 ALM	1,179,648 bit
10GbE (Arria 10)	156.25MHz	1,327 ALM	1,179,648 bit
40GbE (Arria 10)	300MHz	2,678 ALM	1,179,648 bit

UDPxxG-IP core standalone compilation result

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

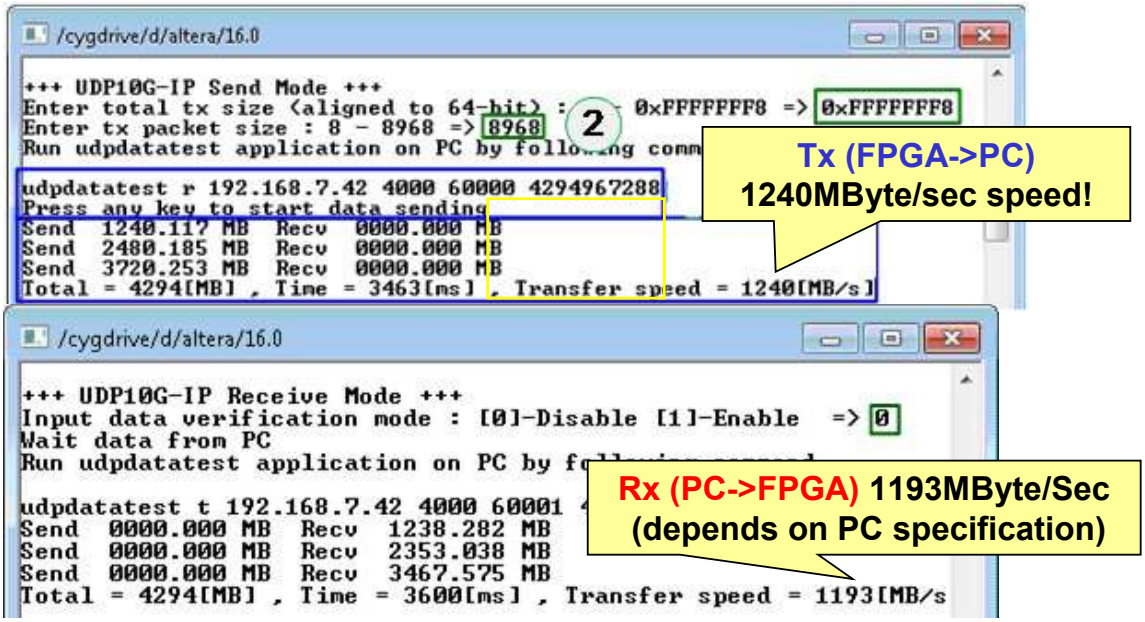
Transfer Performance 1 (GbE)

- Measured real Tx/Rx performance of GbE core



Transfer Performance 2 (10GbE)

- Measured real Tx/Rx performance of 10GbE core



Transfer Performance 3 (40GbE)

- **Measured real Tx/Rx performance of 40GbE core**
  (Xfer result between two FPGA boards including UDP40G-IP core)

```
Send 123843 MByte Recv 0 Byte
Send 128797 MByte Recv 0 Byte
Send 133750 MByte Recv 0 Byte
Send data complete

Total tx transfer size = 4294967295 (256-bit)
Total = 137.438[GB] , Time = 27745[ms] , Transfer speed = 4953[MB/s]
```

UDP40G-IP in Tx side FPGA
4953MByte/sec speed!

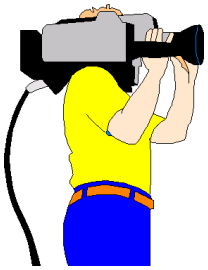
```
Send 0 Byte Recv 125596 MByte
Send 0 Byte Recv 130064 MByte
Send 0 Byte Recv 134532 MByte
Receive data completed

Total rx transfer size = 4294967295 (256-bit)
Total = 137.438[GB] , Time = 27803[ms] , Transfer speed = 4943[MB/s]
```

UDP40G-IP in Rx side FPGA
4943MByte/sec speed!

UDPxxG-IP Application

- **Video-on-Demand via Broadcast**
  - Stream video transmission in real time
  - Requires minimum overhead and latency
  - UDPxxG-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
  - UDPxxG-IP can cover full duplex within minimum latency

For more detail

- **Detailed documents available on the web site**
  - https://dgway.com/UDP-IP_A_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-Jan-2020	English version initial release for all UDPxx-IP series