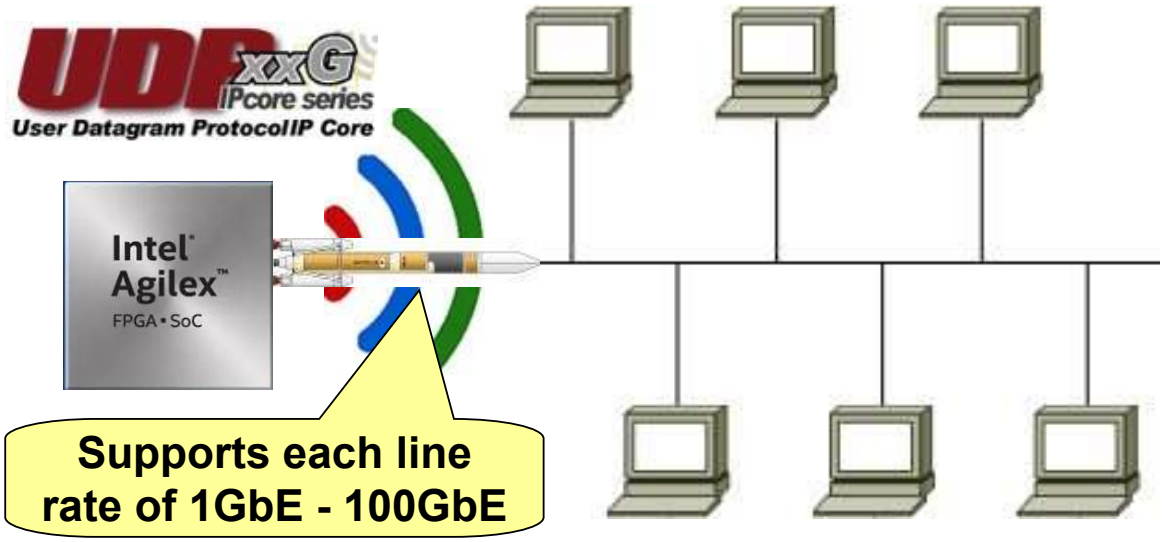
UDPxxG-IP Introduction (Intel)

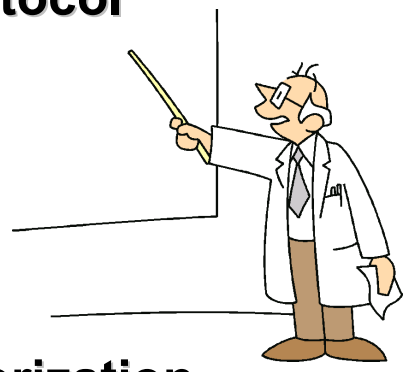
Ver2.0AE



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

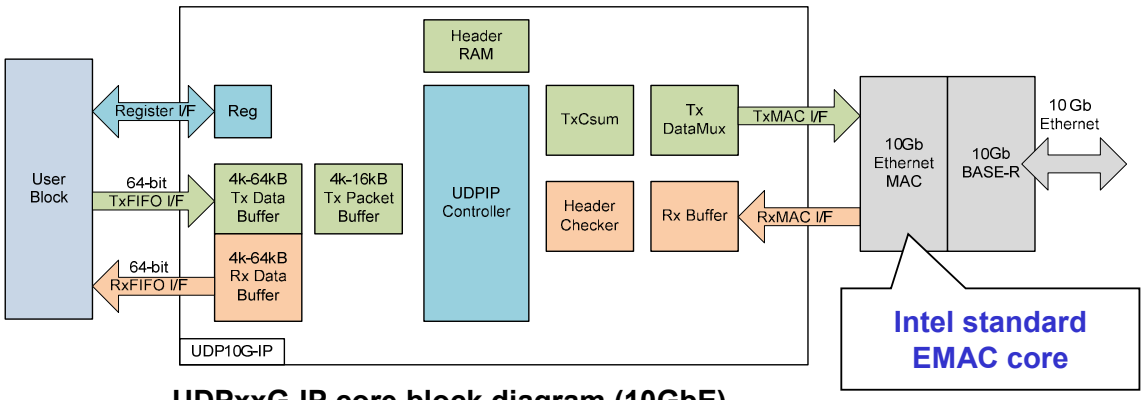
- **Problem in performance and latency**
  - CPU resource consumption by UDP packet building
    - Check-sum calculation
    - Concatenate header and transmit data
  - Bandwidth is not stable due to firmware process

- **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, 25GbE, 40GbE, or 100GbE speed
- Inserts between user logic and Intel EMAC module
- Supports Full Duplex communication



UDPxxG-IP core block diagram (10GbE)

UDPxxG-IP core lineup

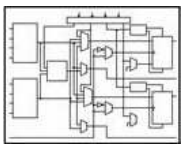
Family | line rate	GbE	10GbE	25GbE	40GbE	100GbE
Cyclone V	Ship OK				
Arria V	Ship OK				
Cyclone 10 GX	Order OK	Order OK			
Arria 10	Ship OK	Ship OK		Ship OK	
Stratix 10	Order OK	Order OK	Order OK	Order OK	Order OK
Agilex F		Order OK	Ship OK	Order OK	Ship OK

Ship OK: Can immediate ship

Order OK: Can place order

UDPxxG-IP core lineup (as of 5th-Aug-2021)

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**

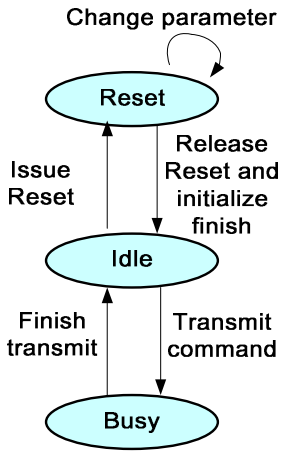
UDPxxG-IP core Advantage 2

- **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

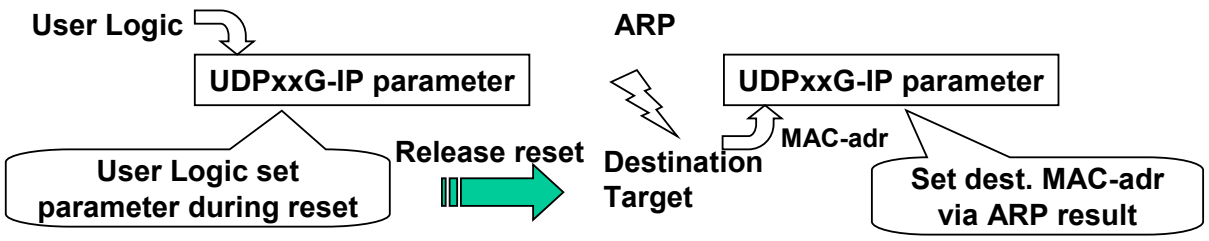
- Set parameter (IP-adr&MAC-adr, etc) during Reset
- Release Reset then initialize including ARP
- Idle state after initialization finish, wait command
- Tx operation starts by user command
- Rx operates at any time except Reset state
  (Accepts all Rx packet if parameter match)
- Tx and Rx operates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



State Diagram

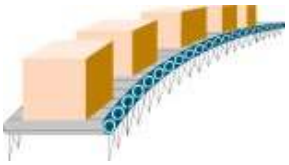
UDPxxG-IP Initialization

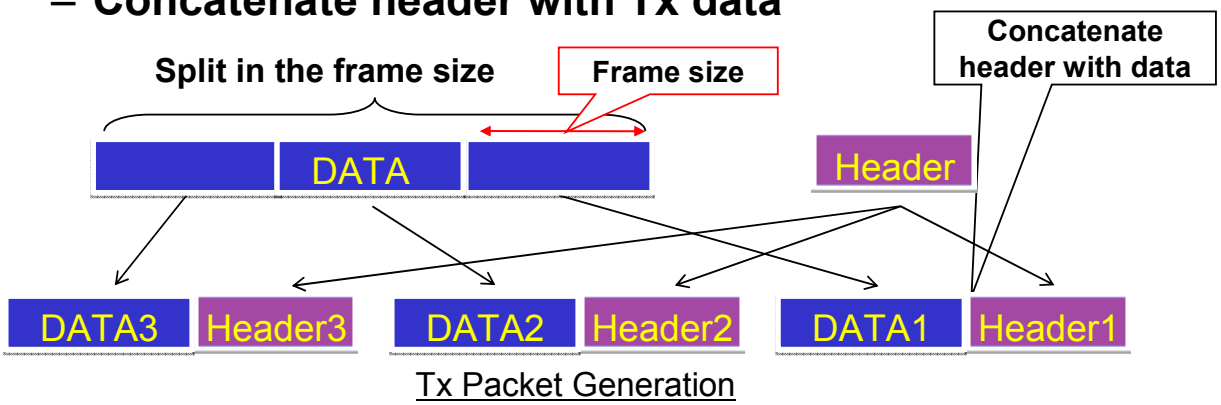
- Set parameter to UDPxxG-IP
  - User logic can set parameter during UDPxxG-IP reset
  - Set IP address, MAC address, and Port number
  - Release reset after parameter setting finish
- UDPxxG-IP executes ARP after reset release
  - Client mode: Issue ARP to the destination target
  - Server mode: Wait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



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

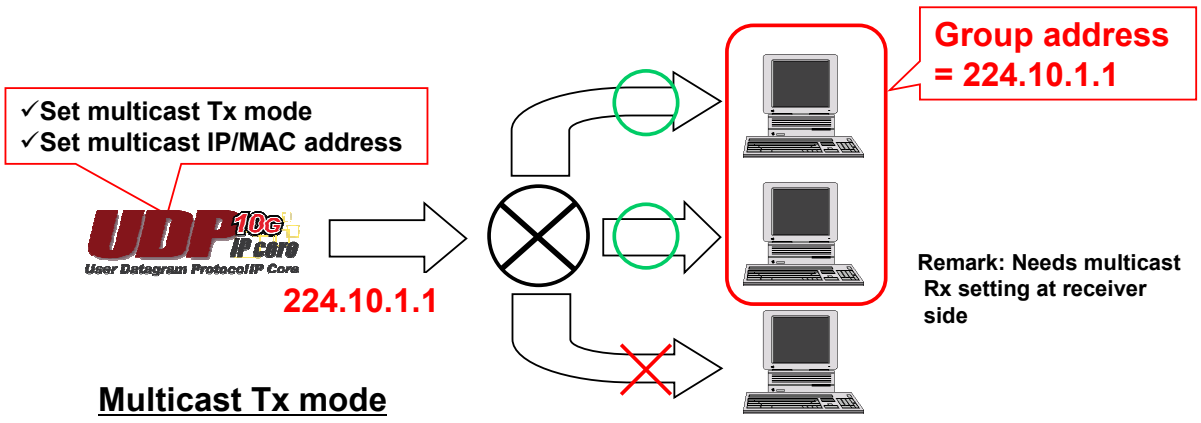




Tx Packet Generation

Multicast/Broadcast High-Speed Tx (optional)

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



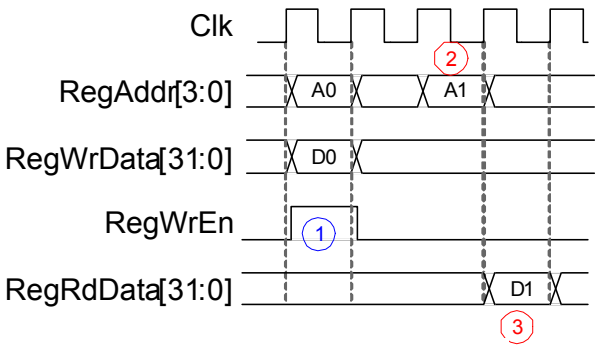
Multicast Tx mode

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

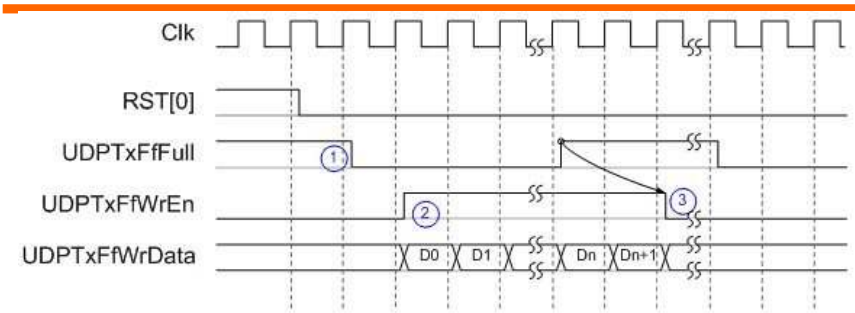


Register I/F timing

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

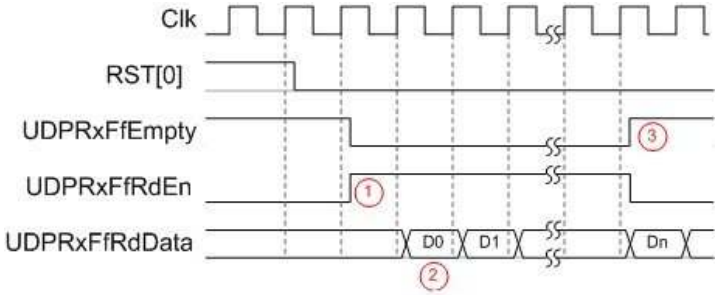
[Register Read]
(2) Set RegAddr
(3) Valid RegRdData output in the next clock

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

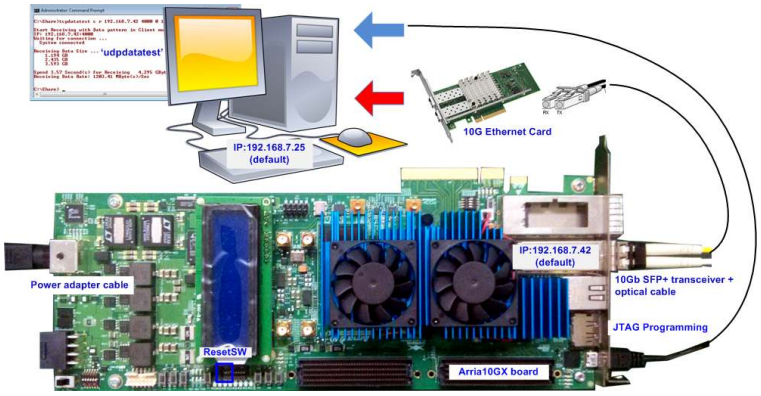
[Rx data read]
(1) Read by RdEn assertion when not Empty
(2) Read data after 1 Clk
(3) Read is inhibited when Empty

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

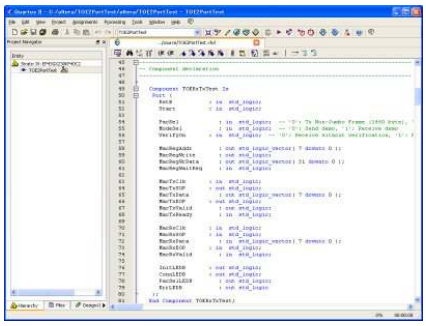
- **SOF file for evaluation with Intel standard board**
  - **Real communication check between FPGA board and PC (Two FPGA boards cross connection for 25/40/100GbE case)**
  - **Measure transfer speed performance and data reliability**



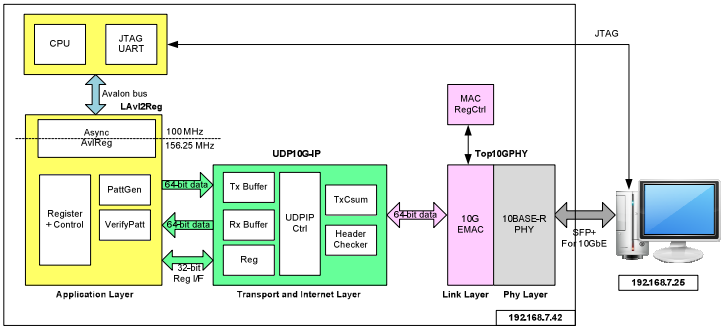
Evaluation environment example using Intel board (Arria10GX)

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



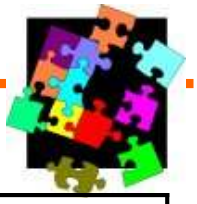
Reference design block diagram

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 (Cyclone V E)	125MHz	1,030 ALMs	1,181,696 bit
10GbE (Arria 10 GX)	156.25MHz	1,347 ALMs	1,179,648 bit
25GbE (Agilex F)	350MHz	1,936 ALMs	1,179,648 bit
40GbE (Arria 10 GX)	275MHz	2,678 ALMs	1,179,648 bit
100GbE (Agilex F)	350MHz	5,769 ALMs	1,837,056 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 (100GbE)

```
+++ UDP100G-IP Send Mode +++
Enter transfer size in byte unit (aligned to 512-bit): 64 - 0xFFFFFFFFFFC0 => 0x1FFFFFFFC0
Enter packet size in byte unit (aligned to 512-bit)  : 64 - 8960          => 8960

Run test application on PC by following command
udpdatatest r 192.168.100.25 60000 4000 137438953408 1

Press any key to start data sending ...
Start data sending
Send 12408 MByte Recv 0 Byte
Send 24817 MByte Recv 0 Byte
Send 37225 MByte Recv 0 Byte

     |          |

Send 111676 MByte Recv 0 Byte
Send 124085 MByte Recv 0 Byte
Send 136493 MByte Recv 0 Byte
Send data complete

Total tx transfer size = 2147483647 512-bit
Total = 137.438[GB] , Time = 11076[ms] , Transfer speed = 12408[MB/s]
```

Sends 128GBytes data to another FPGA by Jumbo Frame (8960byte)

Transfer Performance = 12,400MByte/sec!

Half-Duplex 100GbE transfer result of two FPGAs communication

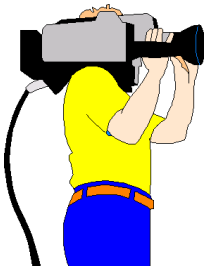
Performance (each line rate)

Line rate	Tx (FPGA->PC)	Rx (PC->FPGA)	condition
1GbE	117MByte/s	110MByte/s	FPGA-PC xfer
10GbE	1,240MByte/s	1,193MByte/s	FPGA-PC xfer
Line rate	Half-duplex	Full-duplex	condition
25GbE	3,097MByte/s	3,067MByte/s	2 FPGA Boards xfer
40GbE	4,953MByte/s	4,925MByte/s	2 FPGA Boards xfer
100GbE	12,400MByte/s	12,343MByte/s	2 FPGA Boards xfer

UDPxxG-IP core performance result (jumbo frame) of each line rate

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
1.1E	23-Jun-2021	UDP25G-IP (25GbE) Release
2.0AE	20-Aug-2021	Complete full line up of 1/10/25/40/100GbE