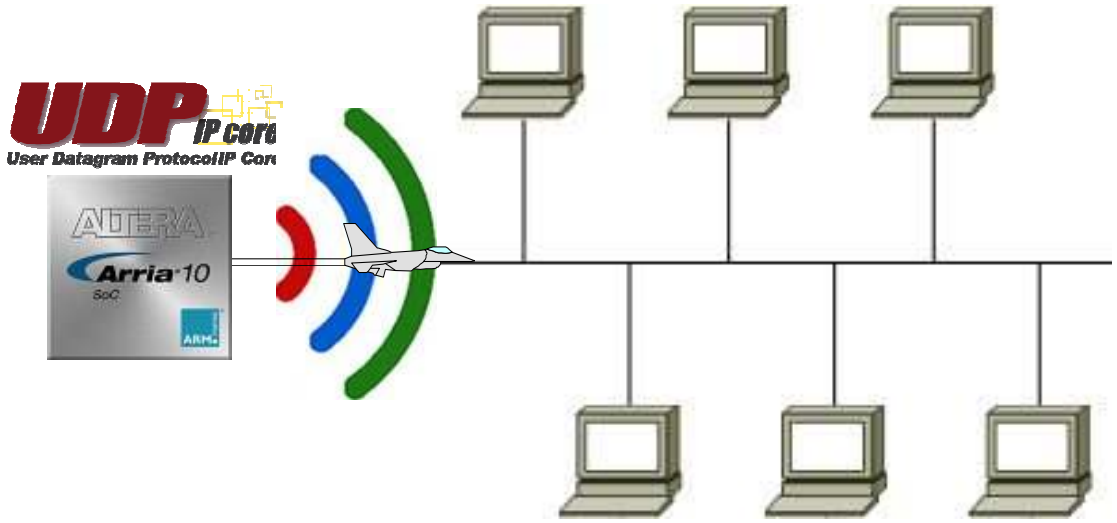
UDP1G-IP Introduction (Intel)

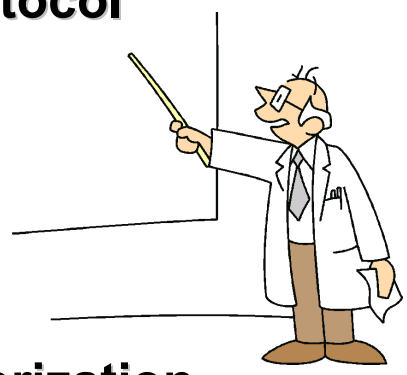
Ver1.01E



Super UDP Speed by hard-wired IP-Core

Agenda

- Merit and demerit of UDP protocol
- UDP1G-IP core overview
- UDP1G-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

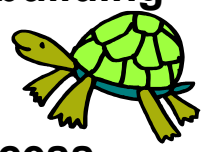


Merit and demerit of UDP protocol

- **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

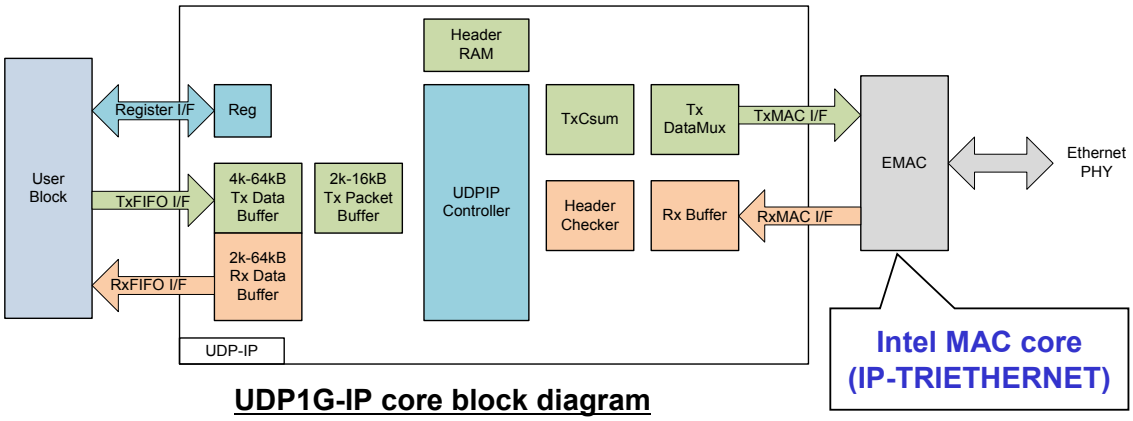
- **Problem in performance and latency**
  - CPU resource consumption by UDP packet building
    - Check-sum calculation
    - Concatenate header and transmit data
  - Bandwidth is not stable due to firmware process
- **The problem gets even worse with full duplex**
  - CPU needs to process time sharing between Tx&Rx
  - Bandwidth and latency further drops
  - Fatal problem for real time application



UDP1G-IP core can provide ideal solution!

UDP1G-IP core Overview

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



UDP1G-IP core block diagram

UDP1G-IP core Advantage 1

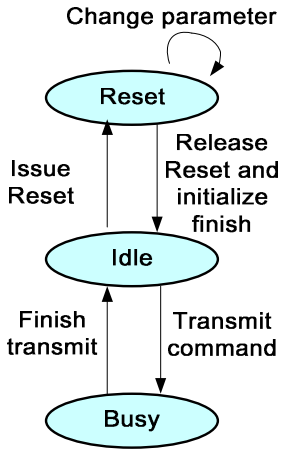
- Fully hard-wired UDP protocol control
  - Possible to build CPU-less network system
  - Zero load for CPU
- Support all of Tx only, Rx only, and full-duplex
  - More than 110MByte/sec real performance
- Can even keep some data reliability
  - Tx: Calculate check sum and build header automatically
  - Rx: Discard received Packet if check sum does not match

UDP1G-IP core Advantage 2

- **Selectable data buffer size**
  - Selectable buffer size of memory usage vs. performance
- **Compatible with Intel MAC core** (IP-TRIETHERNET)
  - Direct connection between UDP1G-IP and MAC
- **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

UDP1G-IP core Operation

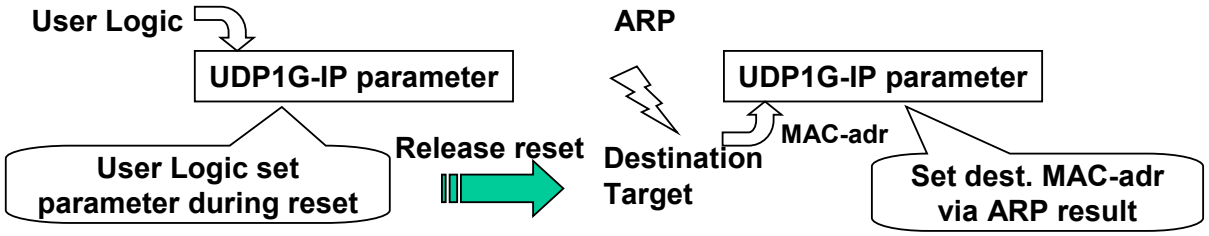
- Set parameter (IP-adr&MAC-adr, etc) during Reset
- Release Reset then initialize including ARP
- Idle state after initialization finish, wait command
- Tx operation starts by user command
- Rx operates at any time except Reset state
  (Accepts all Rx packet if parameter match)
- Tx and Rx operates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



State Diagram

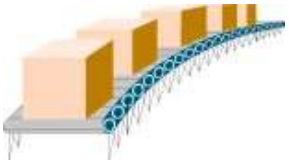
UDP1G-IP Initialization

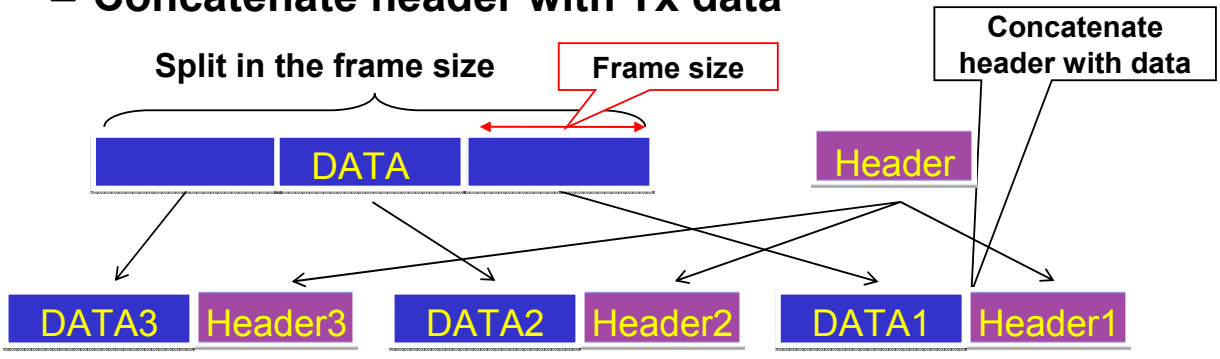
- Set parameter to UDP1G-IP
  - User logic can set parameter during UDP1G-IP reset
  - Set IP address, MAC address, and Port number
  - Release reset after parameter setting finish
- UDP1G-IP executes ARP after reset release
  - Issue ARP to destination target
  - Get MAC-adr of the target via ARP result



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





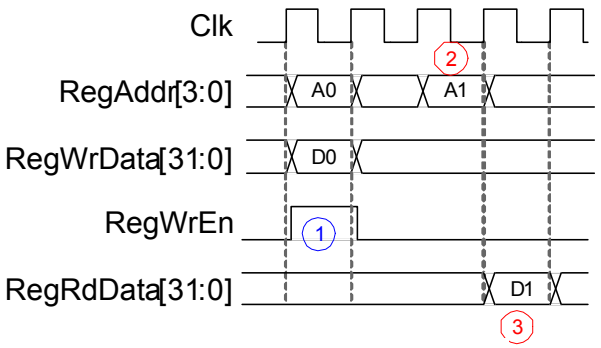
Tx Packet Generation

High-Speed Rx

- **Rx packet header check**
  - Verify all of MAC, IP, and UDP header
  - Supports multicast and broadcast
- **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

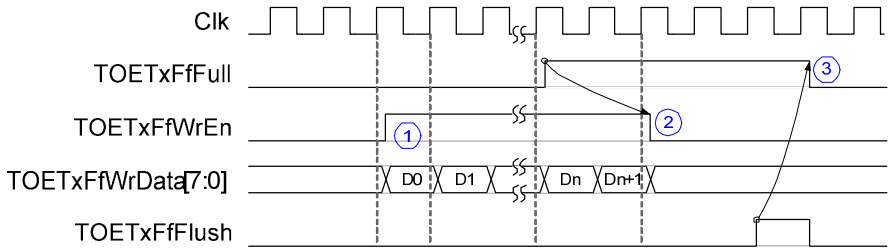


Register I/F timing

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

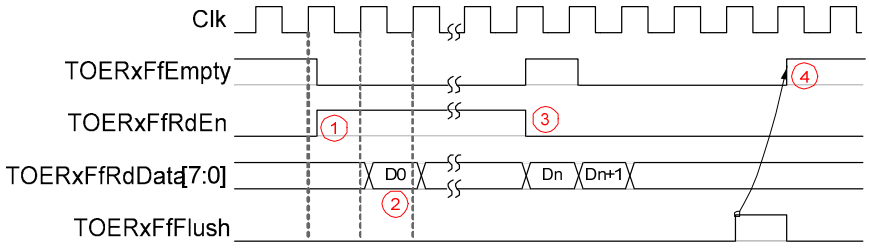
[Register Read]
(2) Set RegAddr
(3) Valid RegRdData output in the next clock

User Interface (Data)



[Tx data write]
(1) Write data with WrEn
(2) Suspend write within 4 Clk after Full assertion
(3) FIFO clear by Flush

Tx 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
(4) FIFO clear by Flush

Rx FIFO I/F timing

Buffer Capacity

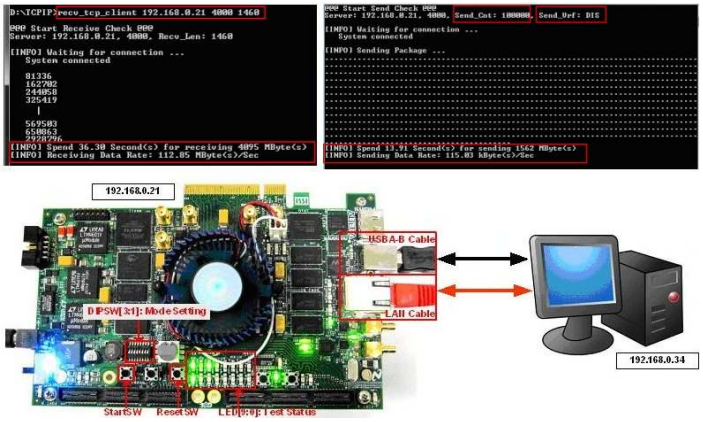
- **Parameterized 3 types of data buffer**
  - (1) Tx Data Buffer: 4KByte - 64KByte
  - (2) Tx Packet Buffer: 2KByte - 16KByte
  - (3) Rx Data Buffer: 2KByte - 64KByte
- **User can optimize resource usage and performance**

Generic Name	Range	Description
TxBufBitWidth	12-16	Set Tx data buffer size in address bit width When set to 12, size is 4KByte, when 16, 64KByte for example.
TxPacBitWidth	11-14	Set Tx packet buffer size in address bit width When set to 11, size is 2KByte, when 14, 16KByte for example
RxBufBitWidth	11-16	Set Rx data buffer size in address bit width When set to 11, size is 2KByte, when 16, 64KByte for example.

Buffer size is selectable by parameterization

Free SOF File for Evaluation

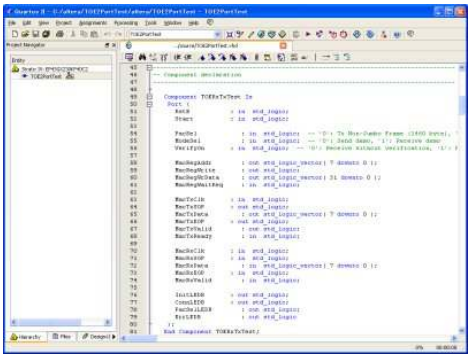
- **SOF file for evaluation with Intel standard board**
  - **Support both Half-Duplex and Full-Duplex operation**
  - **Measure transfer speed performance and data reliability**



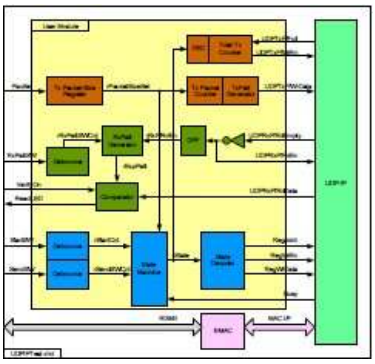
Evaluation environment for Intel board

Reference Design Overview

- **Quartus design project for real operation**
  - **All source code (except IP-core) included in full project**
  - **Both half-duplex and full-duplex design in IP-core package**



Quartus/Qsys project in package



Reference design block diagram

Effective Development on Ref. Design

- Quartus project is attached to UDP1G-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

- UDP1G-IP core standalone resource usage
  - Condition = Maximum buffer setting
    (TxDataBuf=RxDataBuf=64KB, TxPacketBuf=16KB)

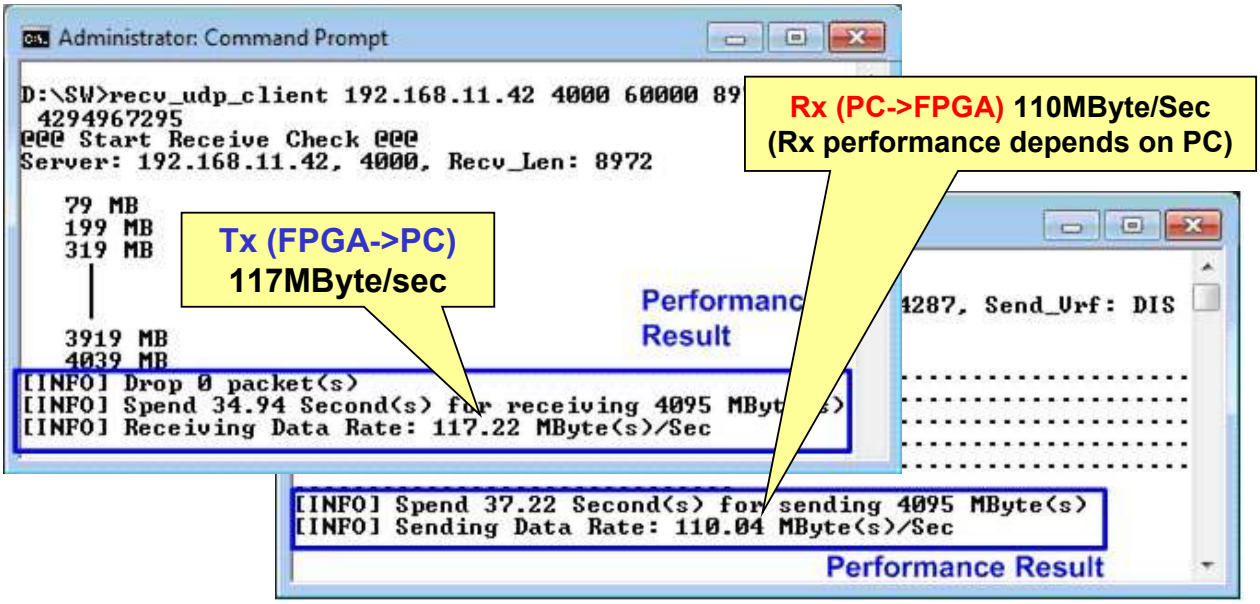
Family	Example Device	Fmax (MHz)	ALMs	Registers	Block Memory bit	Design Tools
StratixIV GX	EP4SGX230KF40C2	125	1,125	1,553	1,181,696	QuartusII 14.0
CycloneV E	5CEFA7F31I7	125	1,048	1,698	1,181,696	QuartusII 15.1
ArriaV GX	5AGXFB3H4F35C5	125	1,047	1,686	1,181,696	QuartusII 14.0
Arria10 SX	10AS066N3F40E2SGE2	125	996	1,635	1,181,696	QuartusII 16.0

UDP1G-IP core standalone compilation result

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

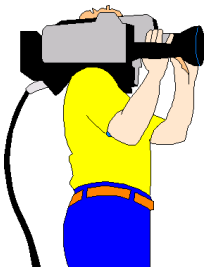
Transfer Performance

- Real performance in data Tx and data Rx



UDP1G-IP Application

- Video-on-Demand via Broadcast
  - Stream video transmission in real time
  - Requires minimum overhead and latency
  - UDP1G-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
  - UDP1G-IP can cover full duplex within minimum latency

For more detail

- **Detailed documents available on the web site**
  - http://www.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	6-Mar-2017	English version initial release
1.01E	7-Dec-2017	Fixed incorrect expression