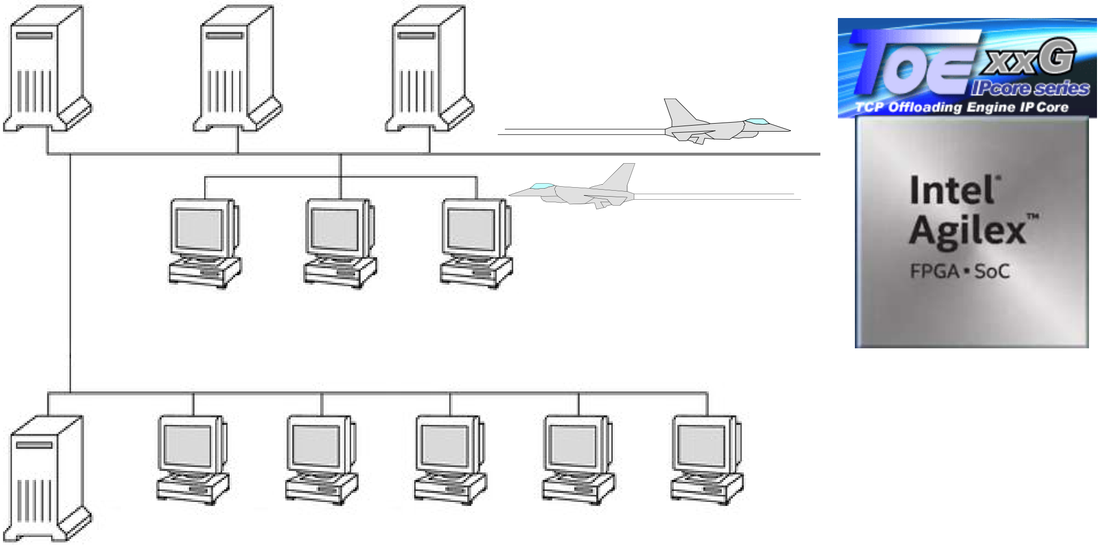
TOExxG-IP Introduction (Intel)

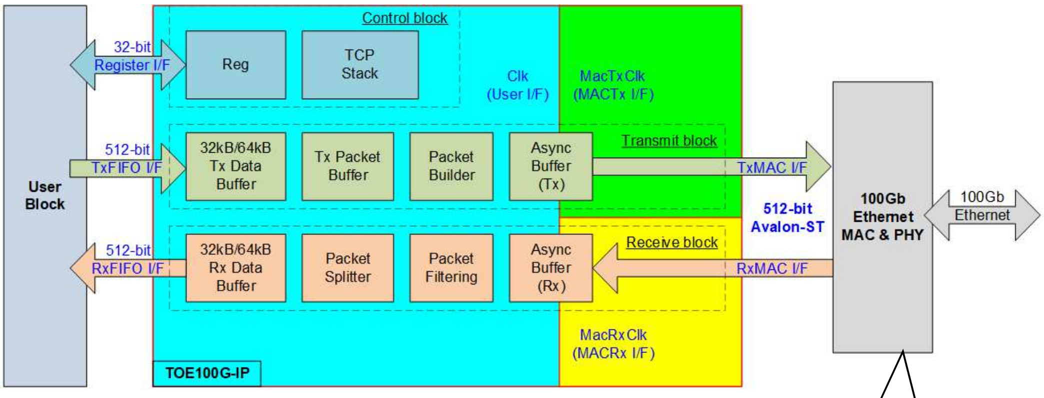
Ver2.0AE



Ultra Hi-Speed TCP by purely hard-wired logic IP-Core!

TOExxG-IP core Overview

- TCP/IP off-loading engine for 100G/40G/25G/10G/1GbE
- Inserts between user logic and Intel EMAC module
- Fully hard-wired TCP control for both Tx and Rx
- Supports Full Duplex communication



TOE100G-IP core (for 100GbE) block diagram

Intel EMAC core

TOExxG-IP core product line up

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

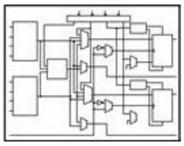
TOExxG-IP core lineup (as of 1st-Aug-2021)

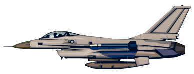
Ship OK: Can immediate ship

Order OK: Can place order

TOExxG-IP core Advantage 1

- Fully hard-wired TCP/IP protocol control
  - Possible to build CPU-less network system
  - Zero load for CPU
- Support all of Tx only, Rx only, and full-duplex
  - 90% real transfer speed of line rate for half-duplex
  - 80% real transfer speed of line rate for full-duplex
- Guarantee transfer data reliability
  - Tx: Automatic retry when No-ACK, Duplicate-ACK, timeout
  - Rx: Automatic ACK control by Sequence number calculation
  - Flow control: Automatic TCP Window Update Packet generation



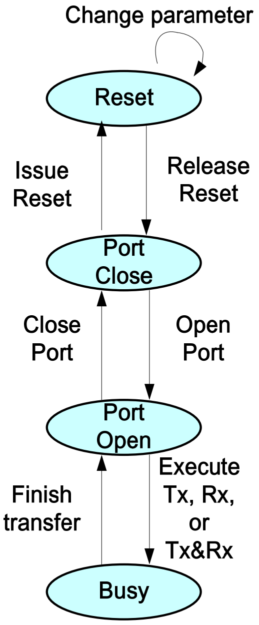


TOExxG-IP core Advantage 2

- Selectable data buffer size
  - Selectable buffer size of memory usage vs. performance
- Compatible with Intel EMAC core
  - Low-cost EMAC-IP core (10GbE) also available
- Many 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

TOExxG-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
- Port open by either of Active (Client) or Passive (Server) mode
- Tx and Rx operates individually (full-duplex)
- If want change parameter, move to Reset state (transfer/packet length can change except Busy)



Change parameter

Reset

Issue Reset

Release Reset

Port Close

Close Port

Open Port

Port Open

Finish transfer

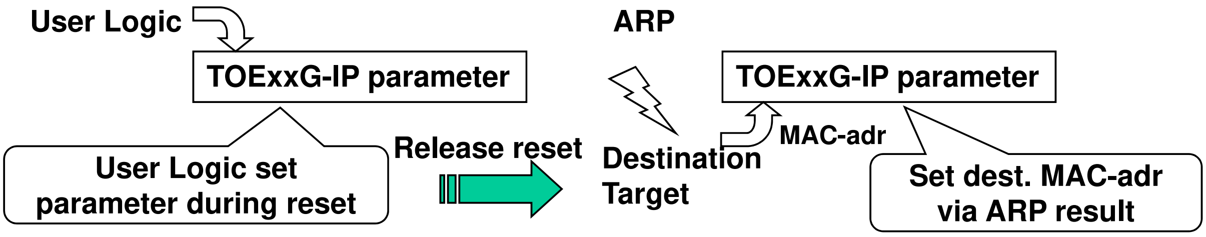
Execute Tx, Rx, or Tx&Rx

Busy

State Diagram

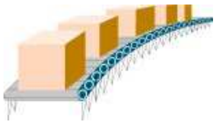
TOExxG-IP Initialization

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

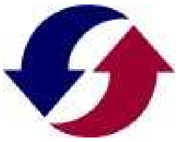


High-Speed Tx

- Tx packet generation
  - User Logic writes Tx data to TxFIFO
  - Split Tx data in the frame size
  - Concatenate header with Tx data

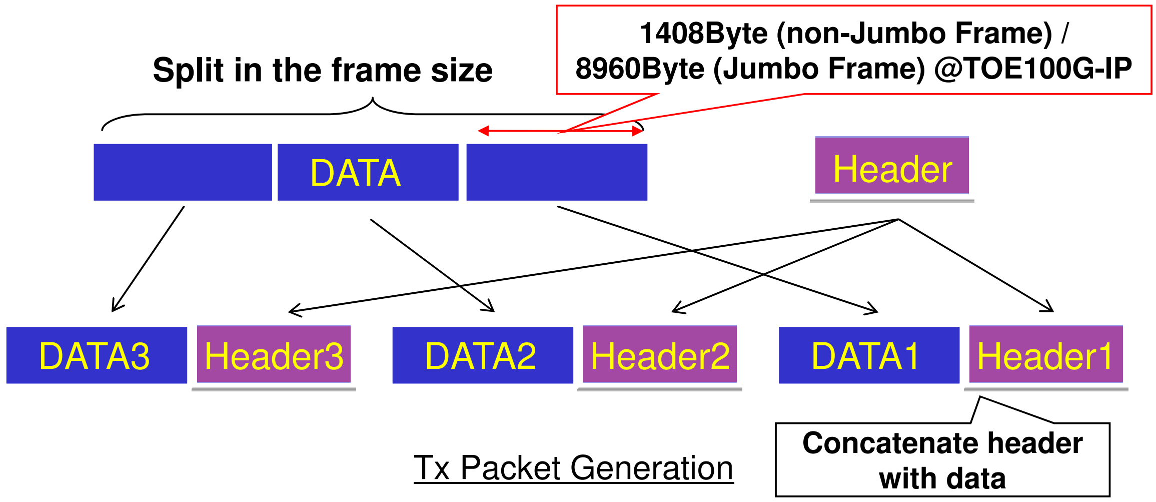


- Automatic retransmit function
  - Check ACK reply from destination
  - Detect No-ACK, Duplicate-ACK, and Timeout
  - Resend same packet by such ACK error detection



Tx Packet Generation

- Generate header and concatenate it with Tx data
  - TOExxG-IP splits Tx data in the frame size
  - Generate checksum and sequence number in TOExxG-IP



Split in the frame size

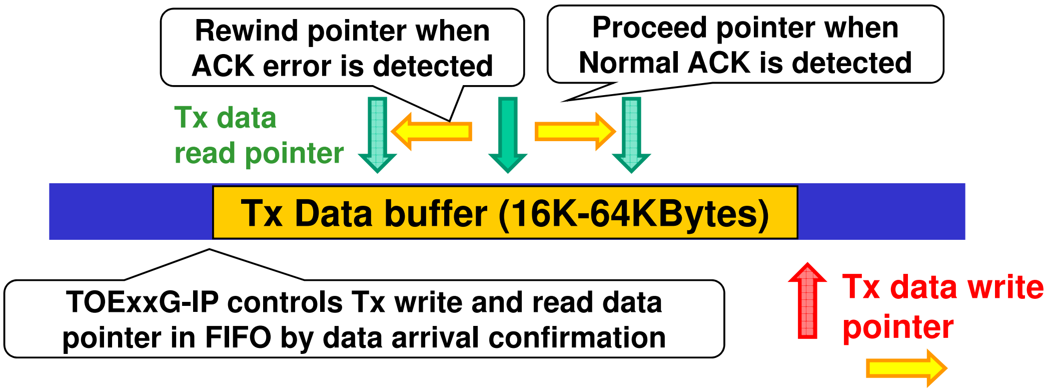
1408Byte (non-Jumbo Frame) / 8960Byte (Jumbo Frame) @TOE100G-IP

Concatenate header with data

Tx Packet Generation

Automatic retransmit

- Retransmit function by dedicated FIFO
  - Proceed pointer by normal ACK reception
  - Rewind pointer by illegal ACK reception
  - TOExxG-IP controls pointer and retransmit operation



Rewind pointer when ACK error is detected

Proceed pointer when Normal ACK is detected

TOExxG-IP controls Tx write and read data pointer in FIFO by data arrival confirmation

High-Speed Rx

- **Rx packet header check**
  - **Ignore packet if destination is not TOExxG-IP or if checksum is wrong**
- **Data reordering**
  - **Reorder when sequence number skip is detected**
  - **Avoid retransmit request for transfer efficiency**
  - **If reordering is not possible, then send duplicate ACK**
- **Duplicate data management**
  - **Check duplicate data in Rx packet**
  - **Retrieve original data by trimming duplicate data part**

Rx Packet Header Check

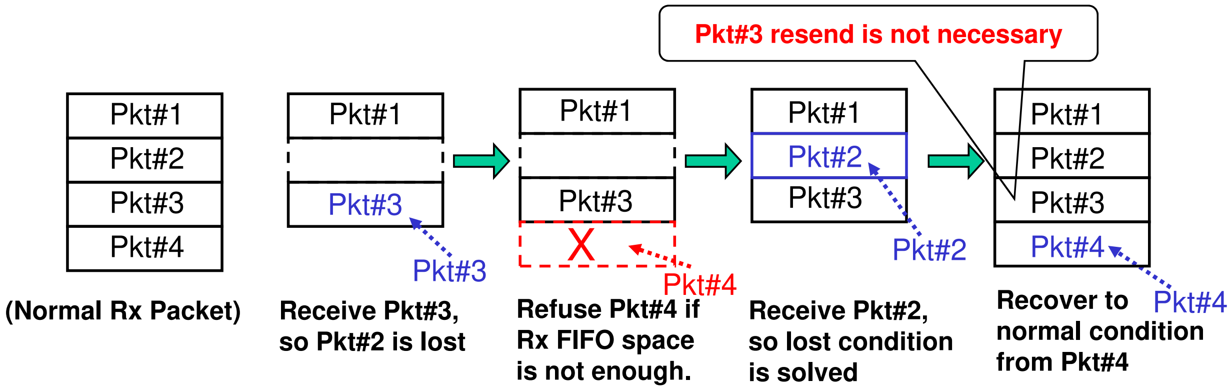
- **Verify header check sum in Rx packet**
  - **Also check following condition in TOExxG-IP**

Byte Offset	Protoco	Description	Check condition
0-5	ICMP	Destination MAC adr	Match with MAC adr set by SML/SMH register
6-11	ICMP	Source MAC adr	Match with target MAC adr set by ARP
12-13	ICMP	Type	= 0x0800 (IP packet)
14	IP	Version/Header	= 0x45 (IPv4, IP header len=20)
20	IP	Flag/Fragment OFS	= b"000000" (no fragment)
23	IP	Protocol Number	= 0x06(TCP packet)
26-29	IP	Source IP adr	Match with IP adr set by DIP register
30-33	IP	Destination IP adr	Match with IP adr set by SIP register
34-35	TCP	Source port number	Match with DPN register or extracted target port number in Passive Open
36-37	TCP	Destination port number	Match with port number set by SPN register
38-41	TCP	Sequence number	Possible value within TOE2-IP core can process this packet

Header check condition in Rx packet

Data Reordering

- Function when SEQ number skip is detected
  - Not accept any packet other than that can solve lost condition
- Data reordering function
  - Recover data contiguous from lost-solved packet
  - Keep performance by suppress resend request



(Normal Rx Packet)

Receive Pkt#3, so Pkt#2 is lost

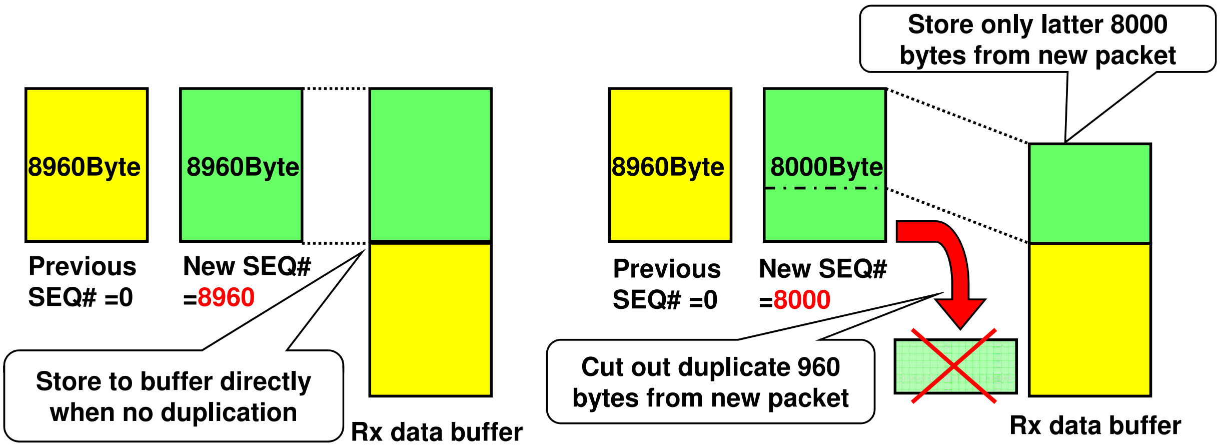
Refuse Pkt#4 if Rx FIFO space is not enough.

Receive Pkt#2, so lost condition is solved

Recover to normal condition from Pkt#4

Duplicate data trimming

- Detect data duplication and correct automatically
  - Detect Rx data duplication by checking sequence number
  - Trim duplicate block and retrieve contiguous data



Previous SEQ# =0

New SEQ# =8960

Store to buffer directly when no duplication

Rx data buffer

Store only latter 8000 bytes from new packet

Previous SEQ# =0

New SEQ# =8000

Cut out duplicate 960 bytes from new packet

Rx data buffer

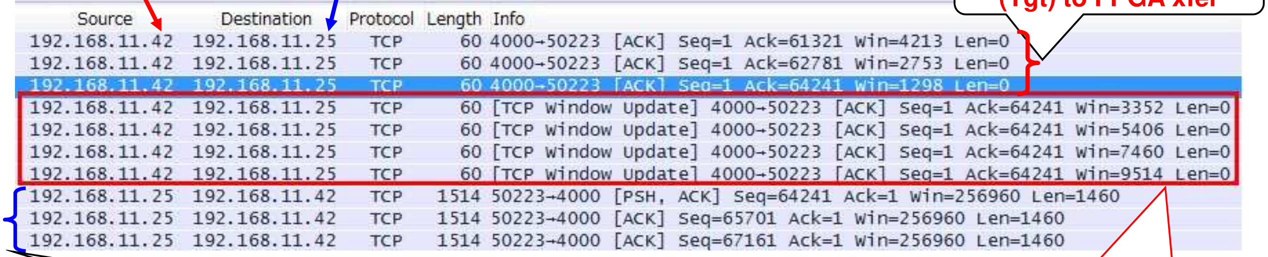
Flow Control (Automatic Window Update packet generation)

- TCP Window Update (ACK) packet generation
  - Detects available space in RX data buffer is restored
  - IP core sends Window Update pkt. at a set threshold.
  - Target side can resume packet sending.

IP: xx.42=FPGA

IP:xx.25=PC (Tgt)

Normal ACK of PC (Tgt) to FPGA xfer



Source	Destination	Protocol	Length	Info
192.168.11.42	192.168.11.25	TCP	60	4000→50223 [ACK] Seq=1 Ack=61321 Win=4213 Len=0
192.168.11.42	192.168.11.25	TCP	60	4000→50223 [ACK] Seq=1 Ack=62781 Win=2753 Len=0
192.168.11.42	192.168.11.25	TCP	60	4000→50223 [ACK] Seq=1 Ack=64241 Win=1298 Len=0
192.168.11.42	192.168.11.25	TCP	60	[TCP Window Update] 4000→50223 [ACK] Seq=1 Ack=64241 Win=3352 Len=0
192.168.11.42	192.168.11.25	TCP	60	[TCP Window Update] 4000→50223 [ACK] Seq=1 Ack=64241 Win=5406 Len=0
192.168.11.42	192.168.11.25	TCP	60	[TCP Window Update] 4000→50223 [ACK] Seq=1 Ack=64241 Win=7460 Len=0
192.168.11.42	192.168.11.25	TCP	60	[TCP Window Update] 4000→50223 [ACK] Seq=1 Ack=64241 Win=9514 Len=0
192.168.11.25	192.168.11.42	TCP	1514	50223→4000 [PSH, ACK] Seq=64241 Ack=1 Win=256960 Len=1460
192.168.11.25	192.168.11.42	TCP	1514	50223→4000 [ACK] Seq=65701 Ack=1 Win=256960 Len=1460
192.168.11.25	192.168.11.42	TCP	1514	50223→4000 [ACK] Seq=67161 Ack=1 Win=256960 Len=1460

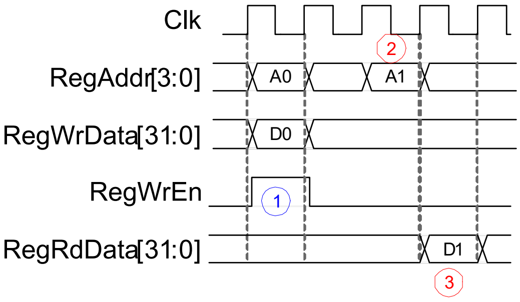
Resume data sending by Window size restore

Automatic Window Update pkt. sending (Example of TOE1G-IP operation)

IP core generates Win. Update pkt. by Rx buffer restore

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

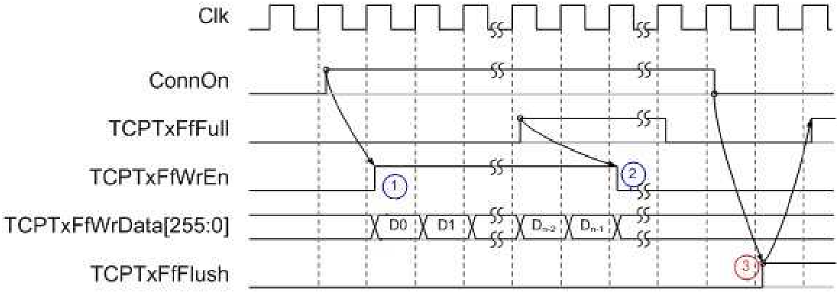


Register I/F timing

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

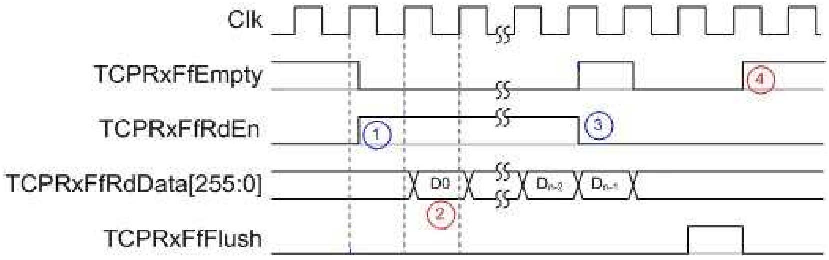
[Register Read]
(2) Set RegAddr
(3) Valid RegRdData output in the next clock

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 2 types of data buffer**
  (1) Tx Data Buffer: 16K/32K/64KBytes
  (2) Rx Data Buffer: 16K/32K/64KBytes
- **User can optimize resource usage and performance**

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

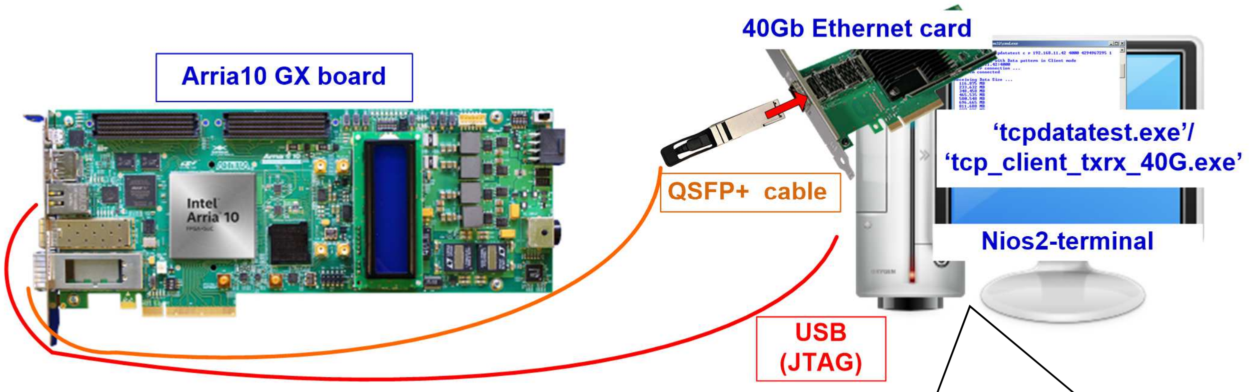
Buffer size is selectable by parameterization

2 types free SOF file for Evaluation

- 2 types free SOF file for evaluation with Intel board
  - Communicate between (A)PC and FPGA, or (B)Two FPGA Bds.
  - For PC-FPGA connection, PC side become bottle neck
    (Real time TCP process by PC with high line rate is not practical)
- Measure transfer performance and data reliability
  - Supports both Half-Duplex and Full-Duplex mode
  - Data reliability check by real time data verification
- Bit file is free of charge
  - User can evaluate IP-core before purchase

Evaluation (A) PC-FPGA

- Use of evaluation software on PC side
  - Note that it can't achieve the best performance due to the bottle neck at PC side

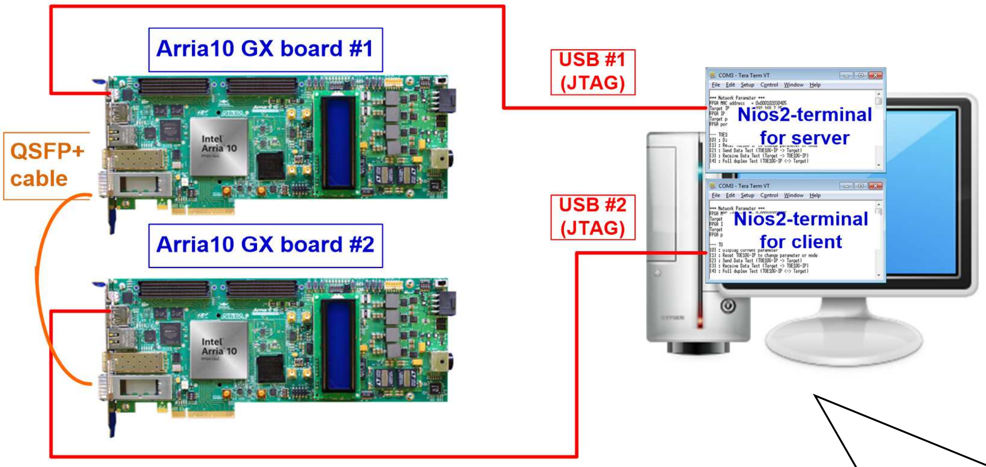


Evaluation (A) PC-FPGA

Remark: Performance is dropped because TCP process at PC is the bottle neck under high line rate data transfer

Evaluation (B) FPGA-FPGA

- Communicate between two FPGA boards
  - External PC controls both two FPGAs via Nios2 console

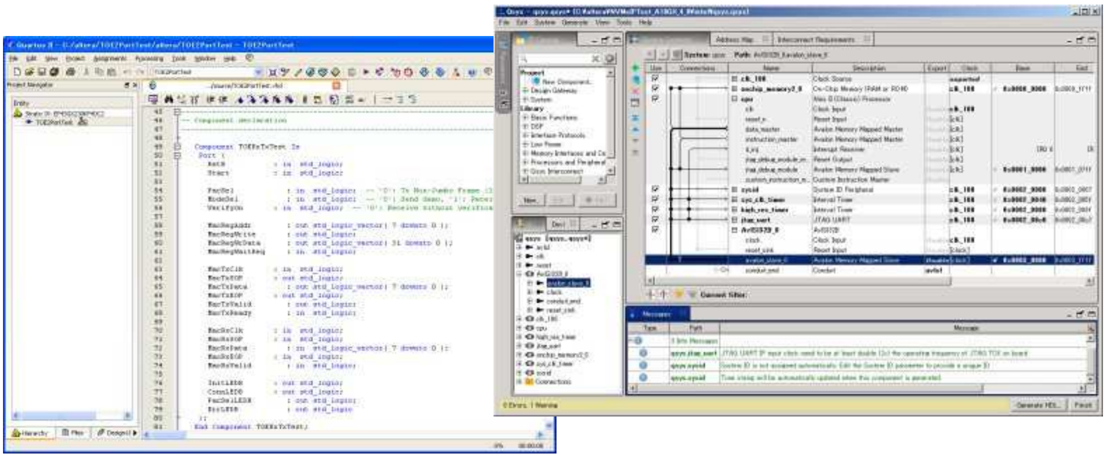


Evaluation (B) FPGA-FPGA

Can achieve the best performance!
TCP data is transferred only between two FPGAs.

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

Effective Development on Ref. Design

- Quartus project is attached to the 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

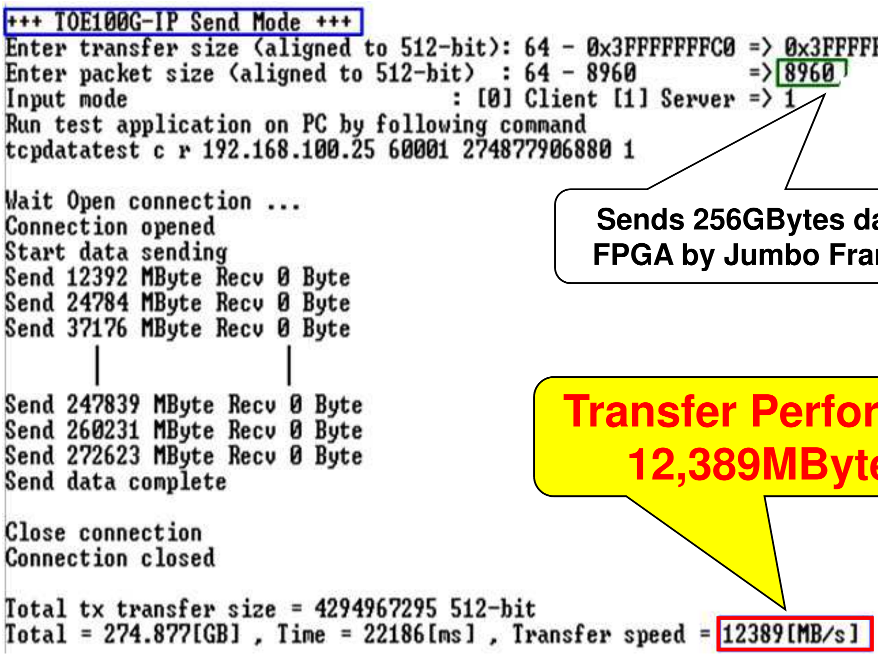
- TOExxG-IP core standalone resource usage
  - Condition = Maximum buffer setting (Tx=Rx=64KB buffer)

Line rate (family)	Clock freq.	Logic usage	Max. memory
GbE (CycloneV E)	125MHz	2047 ALMs	1181696 bit
10GbE (Arria10 GX)	156.25MHz	2453 ALMs	1179648 bit
25GbE (Stratix10 GX)	350MHz	3273 ALMs	1179648 bit
40GbE (Arria10 GX)	322MHz	3656 ALMs	1179648 bit
100GbE (Agilex F)	350MHz	7713 ALMs	1837056 bit

TOExxG-IP core standalone compilation result

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

Performance (100GbE)



```
+++ TOE100G-IP Send Mode +++
Enter transfer size (aligned to 512-bit): 64 - 0x3FFFFFFFC0 => 0x3FFFFFFFC0
Enter packet size (aligned to 512-bit)  : 64 - 8960         => 8960
Input mode                              : [0] Client [1] Server => 1
Run test application on PC by following command
tcpdatatest c r 192.168.100.25 60001 274877906880 1

Wait Open connection ...
Connection opened
Start data sending
Send 12392 MByte Recv 0 Byte
Send 24784 MByte Recv 0 Byte
Send 37176 MByte Recv 0 Byte
       |               |
Send 247839 MByte Recv 0 Byte
Send 260231 MByte Recv 0 Byte
Send 272623 MByte Recv 0 Byte
Send data complete

Close connection
Connection closed

Total tx transfer size = 4294967295 512-bit
Total = 274.877[GB] , Time = 22186[ms] , Transfer speed = 12389[MB/s]
```

Sends 256GBytes data to another FPGA by Jumbo Frame (8960byte)

Transfer Performance = 12,389MByte/sec!

Half-Duplex 100GbE transfer result of two FPGAs communication

Performance (each line rate)

Line rate	Half-duplex	Full-duplex	condition
1GbE	116MByte/s	103MByte/s	FPGA-PC xfer
10GbE	1,200MByte/s	924MByte/s	FPGA-PC xfer
25GbE	3,098MByte/s	3,096MByte/s	2 FPGA Boards xfer
40GbE	4,907MByte/s	4,858MByte/s	2 FPGA Boards xfer
100GbE	12,389MByte/s	11,177MByte/s	2 FPGA Boards xfer

TOExxG-IP core performance result of each line rate

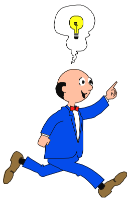
Conclusion

- High Speed data transfer via TOExxG-IP core
  - Practically fastest speed with guaranteed TCP protocol
- Fully hard-wired logic for complicated TCP process
  - Automatic re-send/re-order/duplicated data truncation
- Easy User I/F via register and FIFO
  - Simple enough to build CPU-less system
- Reference design with real operation available
  - Steady development with "design and check" in parallel

TOExxG-IP shall strongly support product development to achieve maximum network use!

For more detail

- Detailed documents available on the web site.
  - https://dgway.com/TOE-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
2.0AE	August 2, 2021	English version for all line up introduction initial release