FPGA based RDMA for BEE Readout

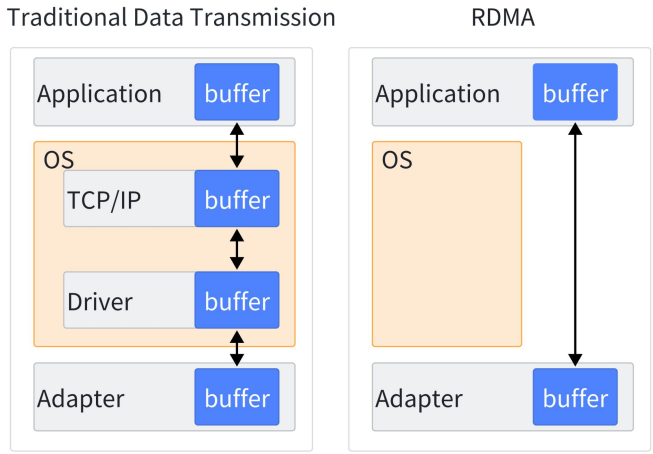
2025.11.09

Outline

- Background
- RoCE v2 on FPGA
- Research Progress
- Summary and Outlook

Background——Data Transmission Protocol

- **TDAQ (Trigger and Data Acquisition System)**
  - Rising luminosity in particle physics
  - Increasing data volumes
  - Critical Needs: Bandwidth & Real-Time Processing
- **Traditional Data Transfer Protocol**
  - I/O bottleneck issue
  - High data copying overhead, limiting data transfer bandwidth
- **RDMA (Remote Direct Memory Access)**
  - Kernel Bypass: Eliminates kernel involvement in the data transmission process
  - Direct data transfer between user-space buffers and NIC buffers
  - High throughput & low latency



Data Transmission progress

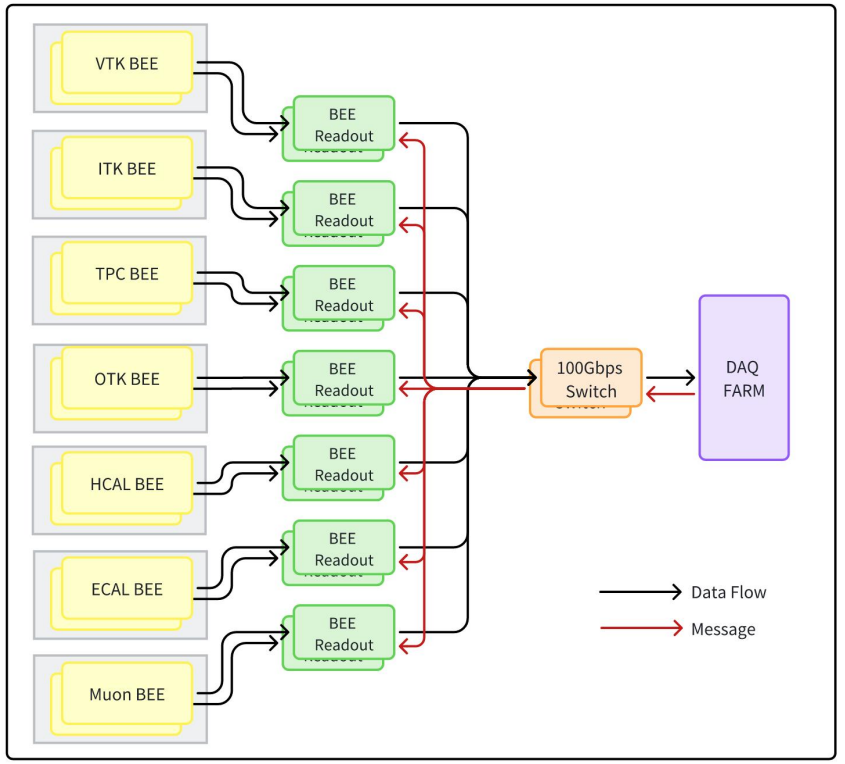
The adoption of RDMA will enable high-performance, efficient data transfer, characterized by high throughput, low latency, and minimal computational overhead.

Data Transmission	RDMA Latency (μs)	TCP/IP Latency (μs)
FPGA - FPGA	0.5-2	10-40
PC - PC	1.5-3	15-50
Best Result	<1	>10

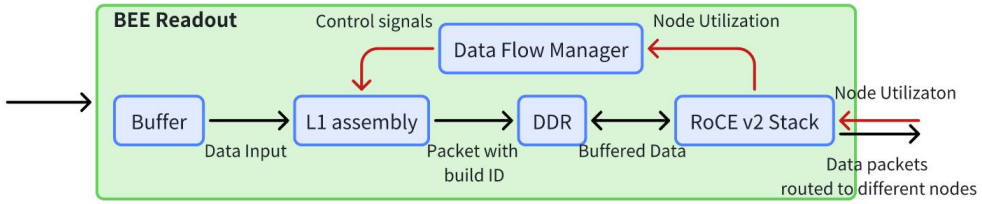
Background——CEPC

- **CEPC**
  - Study the Higgs boson and other physical processes through electron-positron collisions
- **RDMA for BEE Readout**
  - High Data Rate
  - Low-Latency Demand
  - L1 assembly implemented in BEE readout module，enables the dispatch of event data segment to individual HLT nodes
- **Future Work**
  - GPUDirect RDMA: FPGA → GPU Memory
  - Enable faster and more efficient trigger algorithms by cutting latency

- **RDMA network stack on FPGA**
- **BEE readout module**



Data Transmission from BEE to DAQ farm



BEE Readout Module

Background——RDMA Protocol

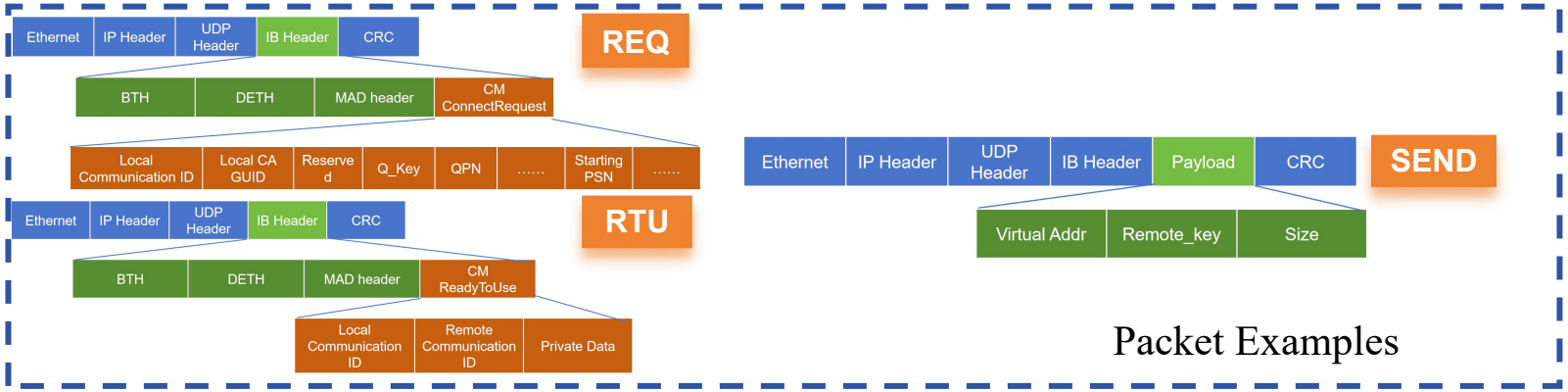
Protocol	Base Transport Protocol	Property	Use Case
InfiniBand	Hardware-based RDMA Protocol	• Low latency、High throughput、Mature CPU offload • Highest cost、WAN complexity	• Event Reconstruction (CMS Run-2) • HPC Cluster • ALICE EPNFarm
RoCE v1	Layer 2 Ethernet	• Leverages Ethernet hardware, low cost • Non-routable	——
RoCE v2	UDP/IP over Layer 3 Ethernet	• Leverages Ethernet hardware, routable, IP-network compatible • A little bit higher latency	• DAQ (ESRF RASHPA、DUNE FPGA) • Event Reconstruction (CMS Run-3，LHCb) • SmartNIC
IWRAP	TCP/IP	• Lowest cost (leveraging mature TCP/IP) • Highest processing overhead and latency	• Specific Cluster Network (CERN)

RoCE v2: A Balanced Choice (Based on complexity, cost, performance, and research trends)

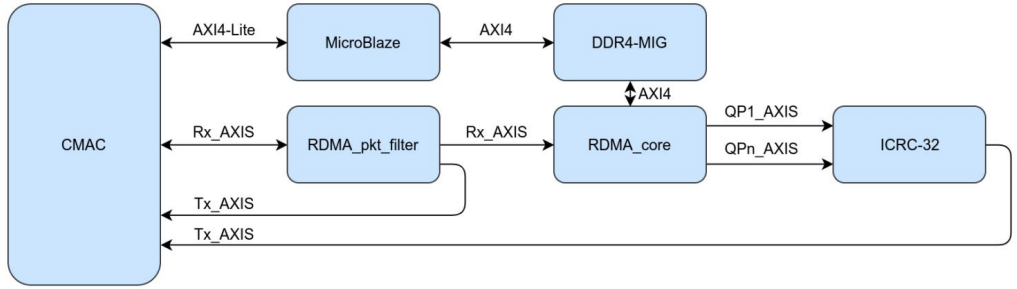
Background——RoCE v2

- **RoCE v2**
  - Connection Setup
    - Manual Out-of-Band (OOB) via TCP/UDP
    - Automated via RDMA CM and librdmacm
  - RoCE v2 Support
    - Operations: Send, Write, Read, Receive…
  - Based on two open source projects

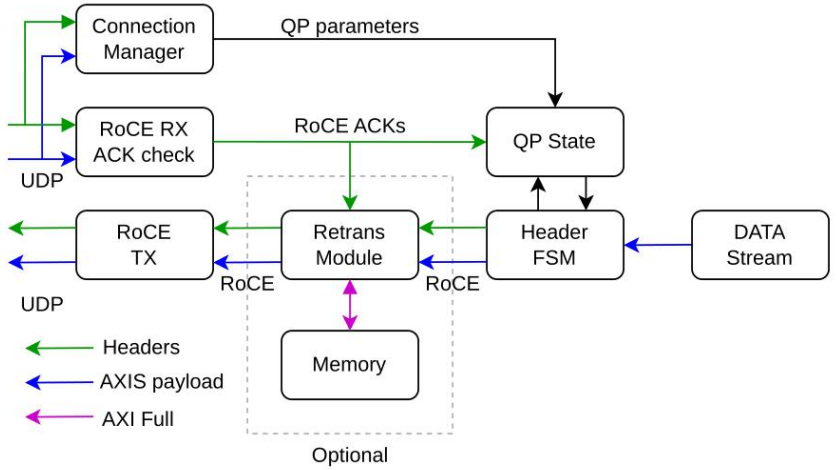
Packet Structure



Packet Examples



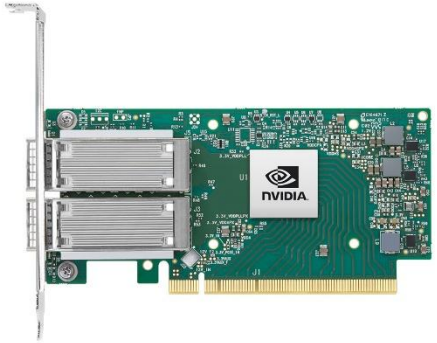
Project 1



Project 2

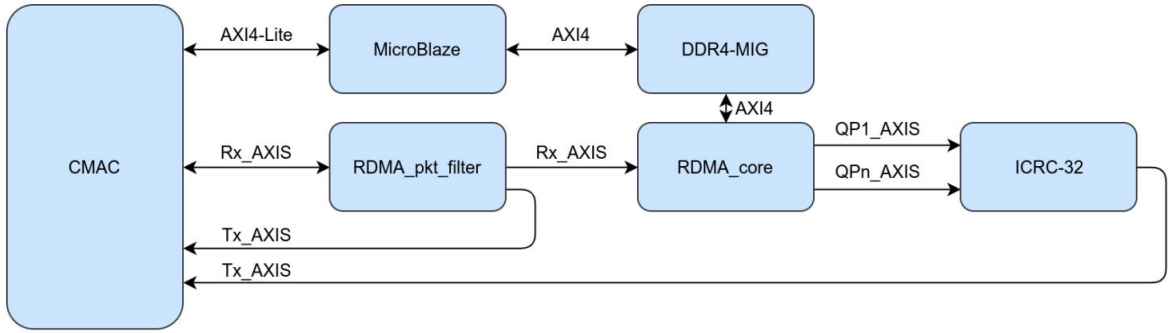
RoCE v2 on FPGA

- **Aims:** HL-LHC FELIX Platform Optimization (ATLAS)
- **Firmware Design**
  - VCU128 Evaluation Kit
  - RDMA core + CMAC + DDR
  - Receive, Send, and Write have been implemented
  - Supports retransmission and multi-QP management



Nvidia Mellanox Connectx-5

AMD FPGA VCU118 Evaluation Kit



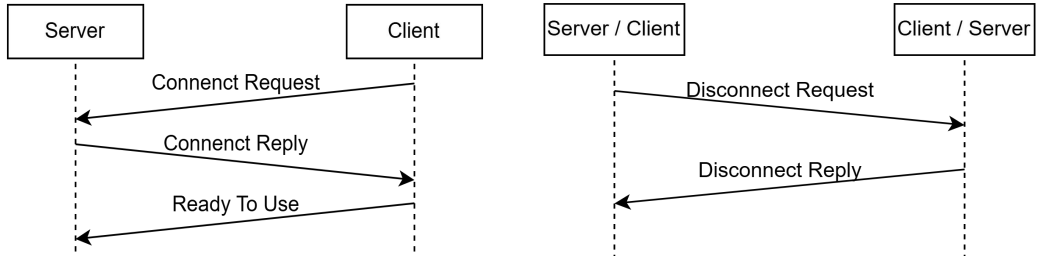
Firmware Design

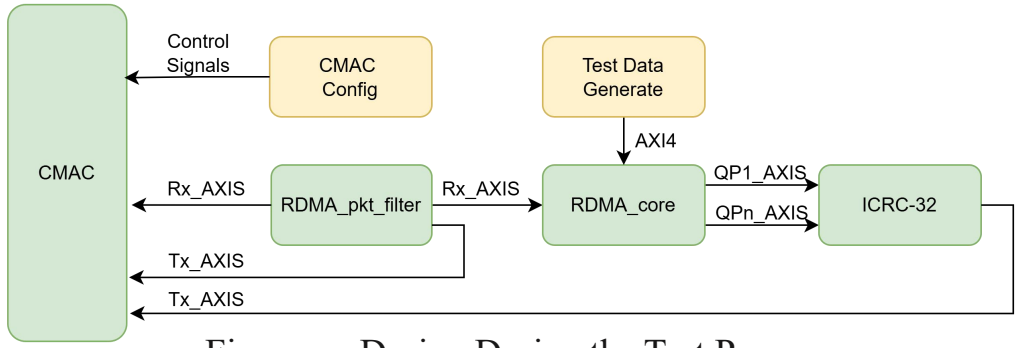
- **RDMA_core:** Parses and generates UDP packets compliant with RoCE v2 protocol
- **RDMA_pkt_filter:** Verifies RoCE v2 packets and responds to ARP messages
- **DDR4 MIG:** Manages temporary data storage with DDR4 SDRAM
- **CMAC:** Frame Assembly, CRC Checking, PCS Interaction

Research Progress——Data transfer process

- **Changes in the Firmware**
  - Fix the timing issue occurred during the transplant process
  - CMAC Config module: Enables configuration of the CMAC core
  - Test Data Generate module: Generate test data

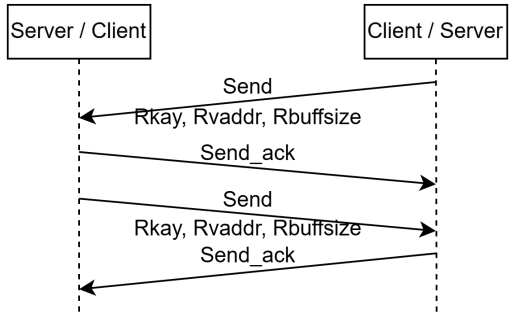
- **Connection Establish**





Firmware Design During the Test Process

- **Send Founction**



- **FPGA-to-PC Data Transmission Successful**
  - **Connection established**
  - **Send function implemented**
  - **Debugging the Write and continuous data transfer functions.**

	PC
CPU	Intel(R) Xeon(R) Gold 6226 CPU @ 2.70GHz
NIC	ConnectX-5
Server/Client	Client

	FPGA board
Model	VCU118 Evaluation Kit
QSFP28	Capable of 100 Gbps Ethernet
Server/Client	Server

RoCEv2 on FPGA

- **Aim:** Replaces TCP/IP to enable direct data transfer from FPGA to Readout Server (CMS)
- **Firmware Design:**
  - Implements RDMA protocol on top of the UDP/IP stack
  - VCU118 Evaluation Kit + ConnectX-5

Implementing RoCEv2 in Verilog for FPGA

Gabriele Bortolato[a,b,c], Matteo Migliorini[c], Andrea Triossi[a,b]

[a]INFN sez. Padova, [b]DFA Padova University, [c]CERN

Finanziato dall'Unione europea NextGenerationEU

Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA

ICSC Centro Nazionale di Ricerca in HPC, Big Data and Quantum Computing

CERN

Università degli Studi di Padova MCCXXII

INFN Istituto Nazionale di Fisica Nucleare

Front-end RDMA over Converged Ethernet, real-time firmware simulation
DOI 10.1088/1748-0221/19/03/C03038
https://github.com/Gabriele-bot/100G-verilog-RoCEv2-lite

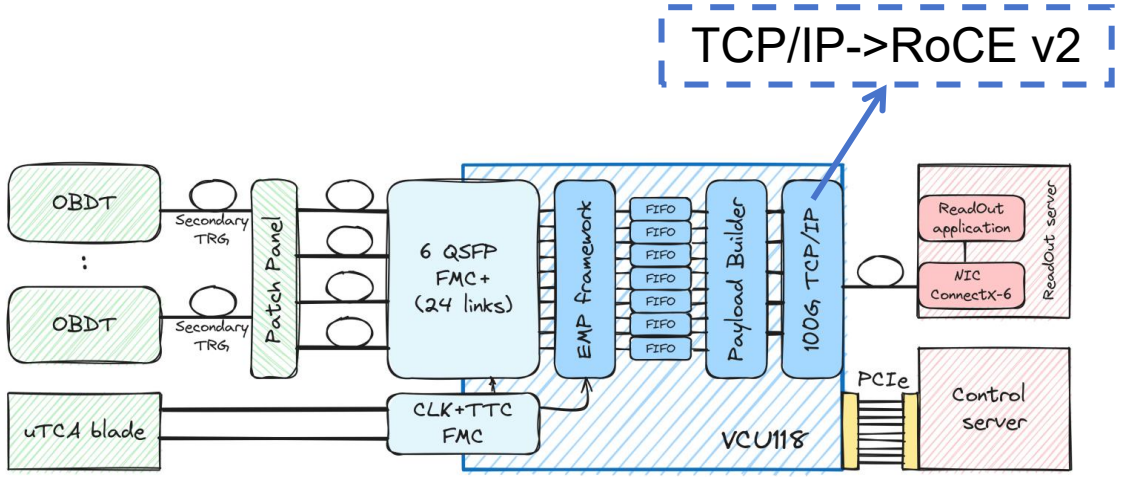


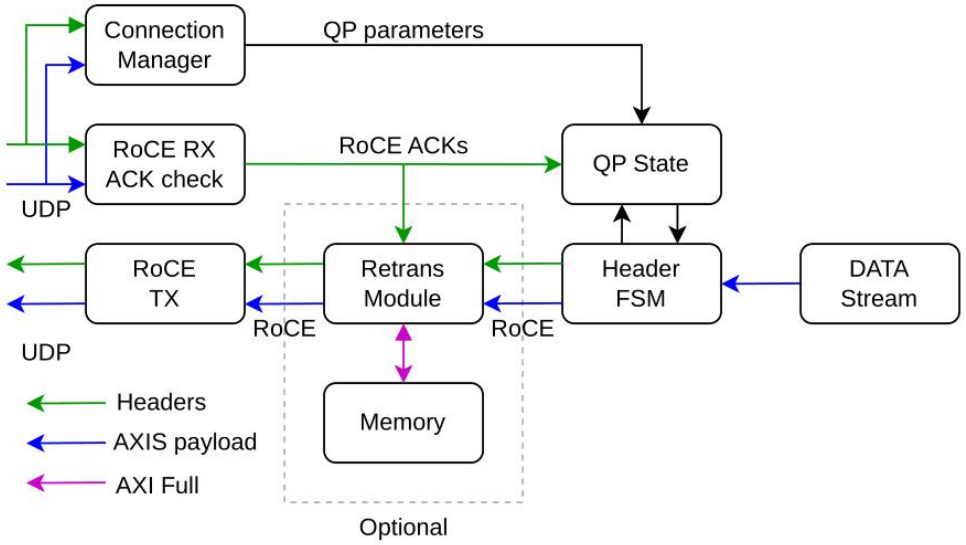
Figure 1. Schematic representation of the DT 40 MHz readout system. From left to right: the connection to the CMS TCDS and secondary lpGBT links from the OBDTs are received by a VCU118. A simplified illustration of its main components is shown in the middle box. The firmware is based on the CMS EMP framework, connected via a set of FIFOs and a payload builder process to a 100G TCP/IP module. Board and firmware are monitored and controlled via PCIe. A second server is used to receive and process the TCP/IP stream.

40 MHz triggerless readout of the CMS Drift Tube muon detector
DOI 10.1088/1748-0221/19/02/C02050

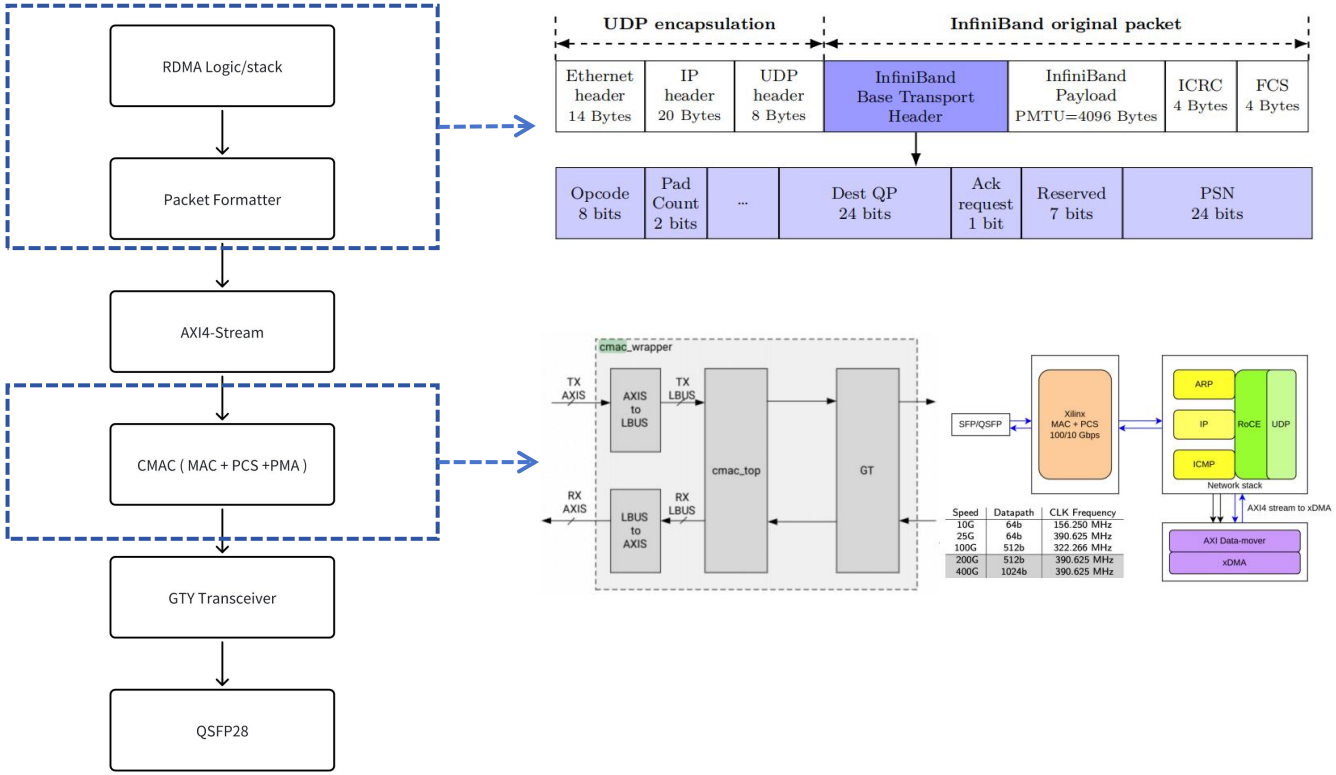
- **Completed project reproduction**
- **Executed firmware functional and performance tests**

Research Progress——Firmware Design

- A transmit-only RoCE v2 implementation
- Supporting only RC RDMA Write and RC Send operations
- The receive path is strictly limited to ACK and NAK processing



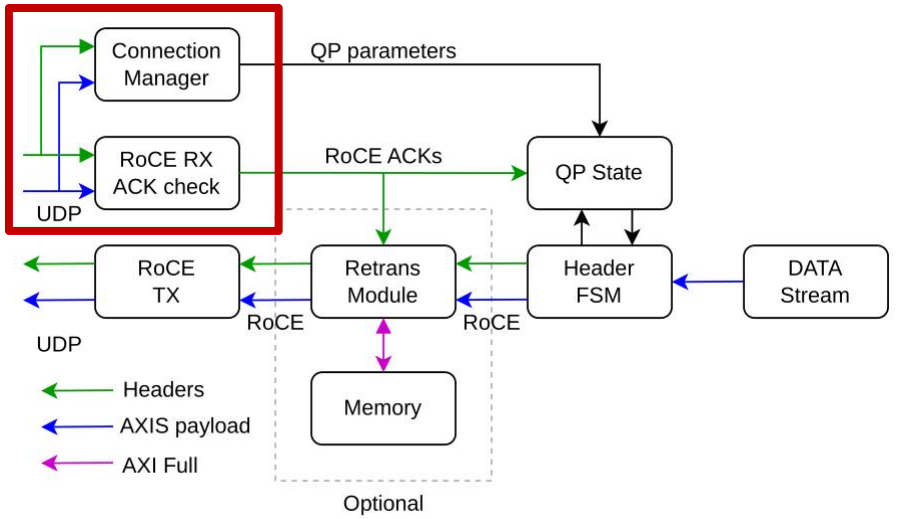
Firmware Design



Data Processing Flow

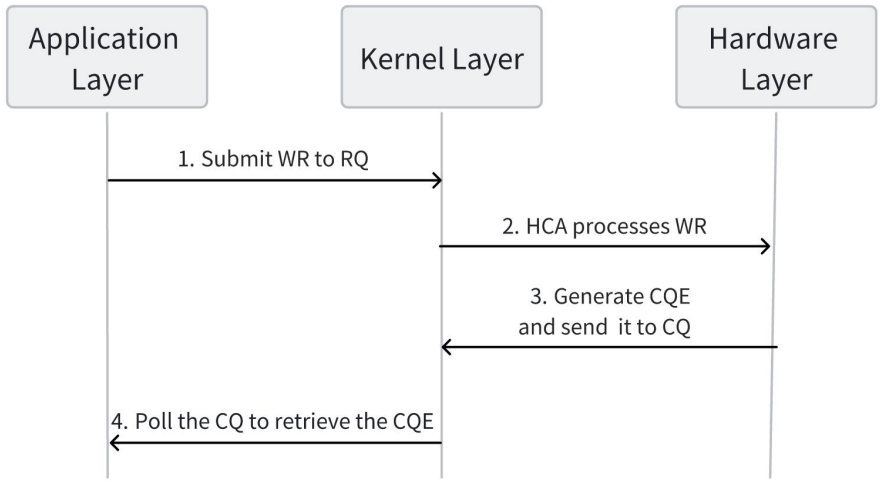
Research Progress——Software Design

- **Server (PC) QP establish**
  1. Open the RDMA device and allocate a Protection Domain (PD)
  2. Allocate a Completion Queue (CQ)
  3. Create a Queue Pair (QP)
  4. Register a Memory Region (MR)
  5. Initialize QP state: RESET → INIT, INIT → RTR, RTR → RTS



- **Synchronize QP information and control the FPGA's QP state**
  - Send connection information (QPN, PSN, MEM Base ADDR, MEM Key) in a fixed frame format to a UDP port other than 4791
  - Transition the FPGA's QP to RTS state via the Connection Manager

Research Progress——Software Design



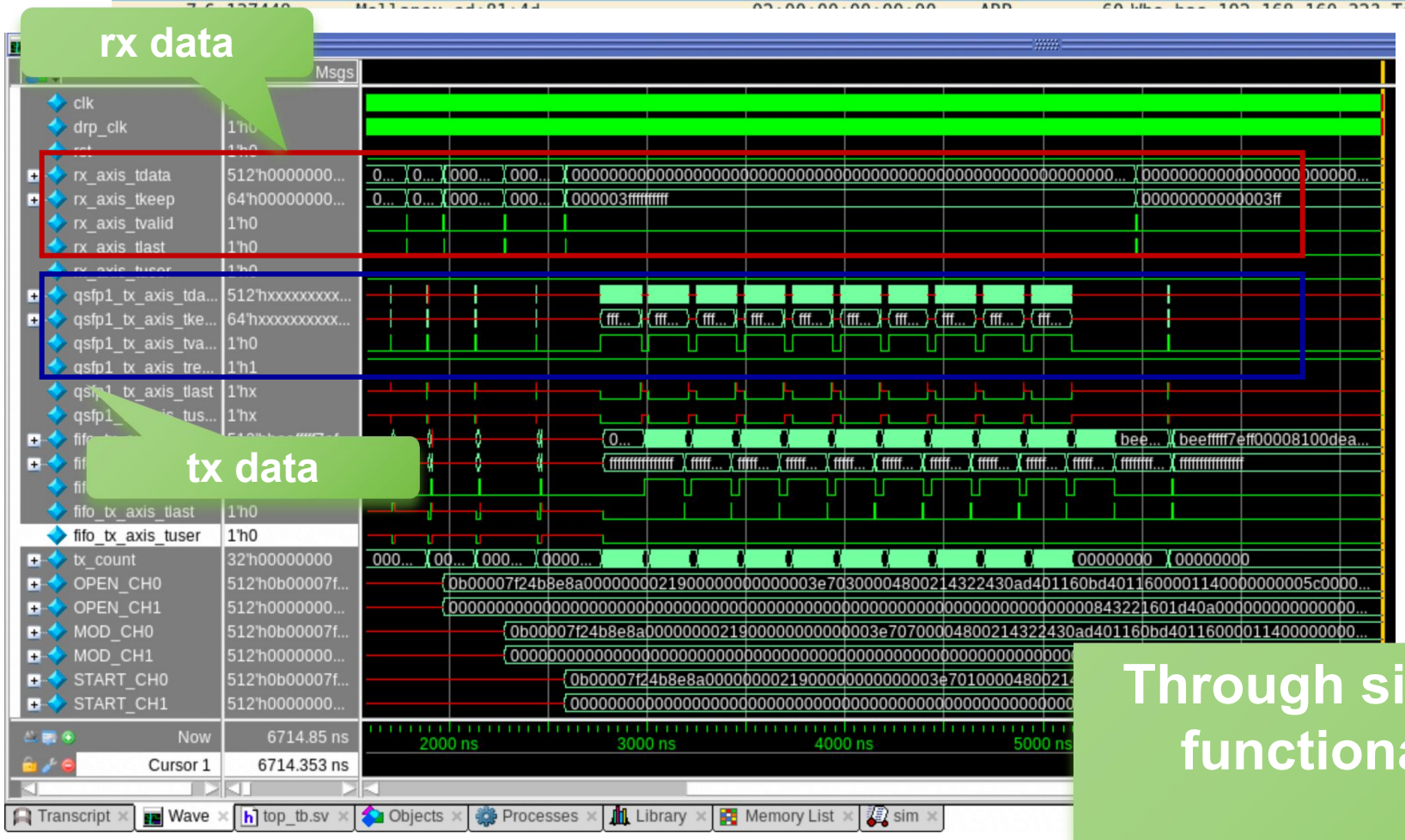
RDMA Workflow on PC

- Post Work Request (WR)
- Hardware processes WR
- Generate Completion Queue Entry (CQE)
- CQE enters Completion Queue (CQ)
- Poll Completion Queue (CQ)

- **Busy-Polling** - Fastest, but CPU intensive
  The application, in a tight loop, continuously and repeatedly calls the polling function (e.g., ibv_poll_cq) to check for new Completion Queue Entries (CQEs) in the Completion Queue (CQ).
- **Event-Driven / Blocking** - Most efficient, but higher latency
  The application does not actively poll. Instead, it requests the hardware to notify it via an "event" when a new CQE arrives. The application's thread "sleeps" or blocks until the event occurs.
- **Hybrid Polling** - A best-practice compromise
  Combines the advantages of the two methods above. It first performs a short period of busy-polling. If no CQE is received during this time, it switches to the event-driven blocking mode, waiting for the next notification.

Functional Test

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.160.32	192.168.160.10	RRoCE	74	RC Send Only Immediate QP=0x0009a6
2	0.000004	192.168.160.10	192.168.160.32	UDP	106	44226 → 17185 Len=64
3	0.000004	192.168.160.10	192.168.160.32	RRoCE	62	RC Acknowledge QP=0x000100
4	4.702509	192.168.160.10	192.168.160.32	ICMP	98	Echo (ping) request id=0x00f5, seq=1/256, ttl=64 (no response found!)
5	5.113446	Mellanox_ed:81:4d	02:00:00:00:00:00	ARP	60	Who has 192.168.160.32? Tell 192.168.160.10
6	5.753488	192.168.160.10	192.168.160.32	ICMP	98	Echo (ping) request id=0x00f5, seq=2/512, ttl=64 (no response found!)

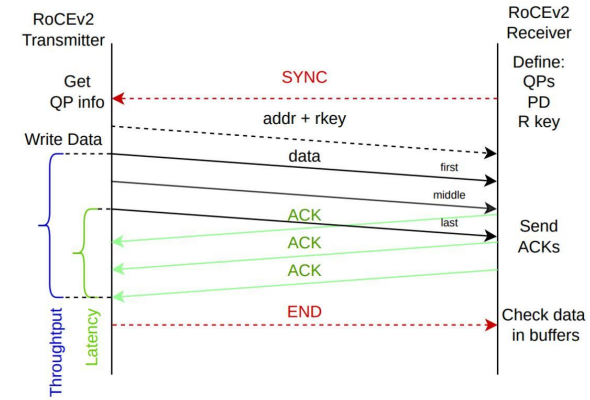


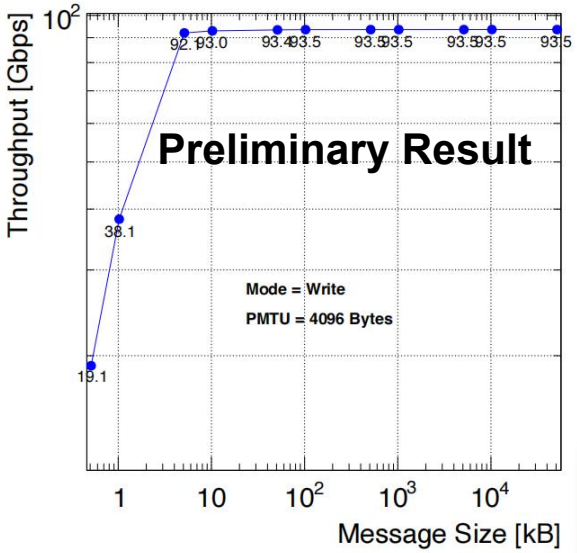
- RDMA Packages Captured
- Simulation (AXI bus data, ping/arp/udp) & Messages
- Fix iCRC padding issue in write function

Through simulation test, the firmware functionality has been verified as correct.

Performance Test

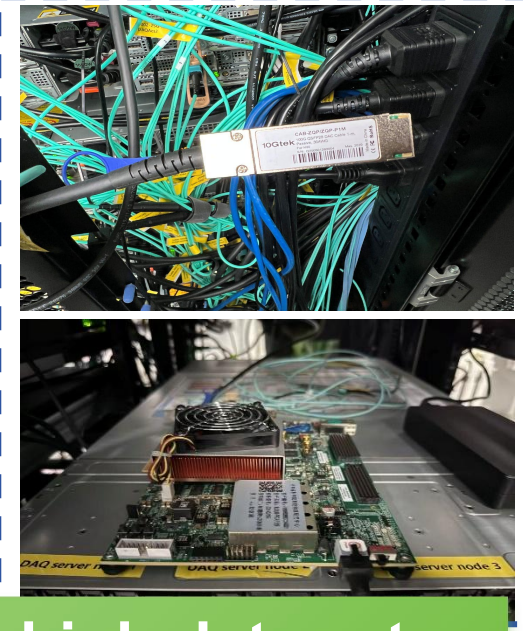
- **Throughput**
  - When the message size exceeds **5 kB**, the attained throughput is **92.1 Gbps**, and the maximum throughput is approximately **93.5 Gbps**
- **Latency**
  - Use OP code, PSN, bth_valid, and aeth_valid to identify the first and last sent packets and validate the received ACK
  - Msg.size=262kB, Msg.num=2e5(50GB)
  - Single package latency ≈ 730 clock ≈ **2.27 μs**





Transmission process demonstration (Multi packages).
Latency of first package = T_first_ack – T_first_write
Latency of last package = T_last_ack – T_last_write
Tot of transfer = T_last_ack – T_start

core_inst/RoCE_minimal_stack_512_instance/transfer_time_tot[63:0]	[U] 1378101107	Input	hw_vio_4
core_inst/RoCE_minimal_stack_512_instance/latency_first_packet[63:0]	[U] 728	Input	hw_vio_4
core_inst/RoCE_minimal_stack_512_instance/latency_last_packet[63:0]	[U] 727	Input	hw_vio_4
core_inst/RoCE_minimal_stack_512_instance/last_acked_psn[23:0]	[U] 15359999	Input	hw_vio_4
core_inst/RoCE_minimal_stack_512_instance/last_buffered_psn[23:0]	[U] 15359999	Input	hw_vio_4



Enabling future software triggers and high data-rate operations under high luminosity.

Performance Test

- **Reliability**
  - Non-PRBS data
    - Test completed successfully
  - Data from Parallel PRBS Generator
    - Parallel PRBS Generator Module
    - Software was developed to verify the consistency between the sent and received data
    - Excessive synthesis time & memory usage (>100 GB during synthesis)

To be continued…

Summary and Outlook

- **Project 1:** Partial functional testing has been completed, and further development is required
- **Project 2:** Developed and tested throughput, latency, and stability during data transmission
  - ✓ Continuous transmission logic
  - ✓ Receiver software development
  - ✓ Throughput (firmware debugging)
  - ✓ 92.1 Gbps throughput, 2.27 μs latency
- **Outlook**
  - **RDMA Network Stack**
    - DDR-Based retransmission implementation
    - Multi-QP Management
    - Priority Flow Control (PFC)
    - Backend computation/Storage pressure
  - **BEE readout module:** The integration with HLT demo will be completed

Thanks!