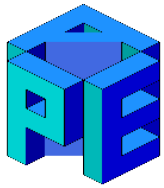



NaNet: a network interface card family for GPU-based real-time systems

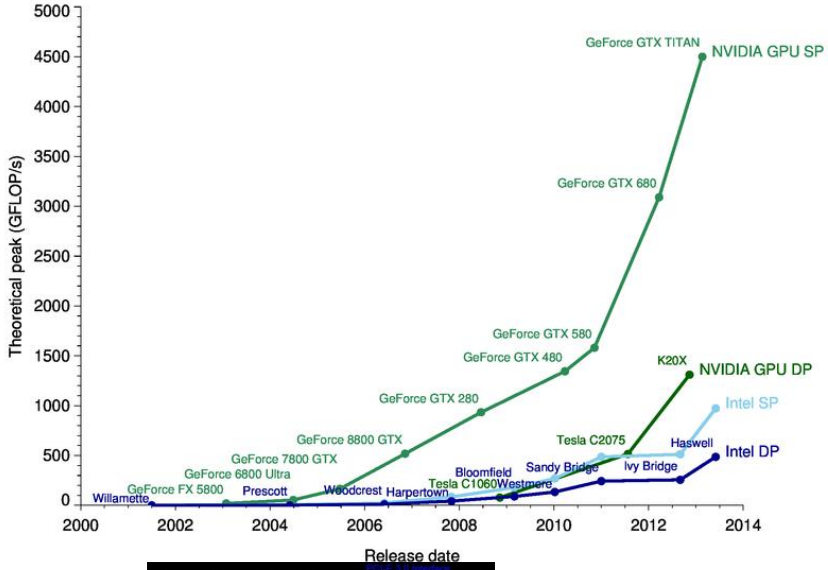
Piero Vicini
(INFN Roma 1)
On behalf of the NaNet collaboration

Super Massive Computation in Theoretical Physics

Trento, February 11-13 2015

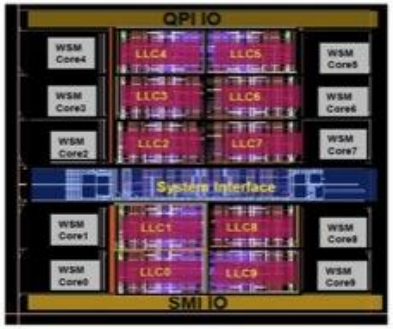
General-Purpose computing on Graphics Processing Units.

- Many-core SPMD architecture (thousands simple cores).
- Several Tflops per single device (e.g. nVIDIA K40: 4.2 SP, 1.4 DP)
- High Memory Bandwidth.
- Power efficient and cost effective
- Higher performance scaling than traditional CPUs.

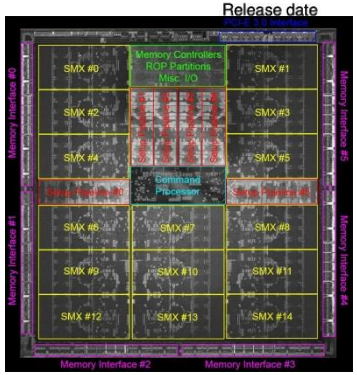


Intel Xeon

- Optimized for latency
- Big Caches



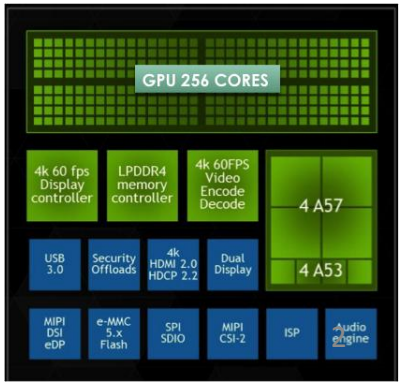
- Optimized for throughput
- Many simple computing units



nVIDIA GK110

- Embedded Hybrid Arch: **Tegra**
  - multicore ARM + (many) GPU cores
  - Low cost, low power, high flops/watt ratio

Tegra X1



GPU in HEP experiments

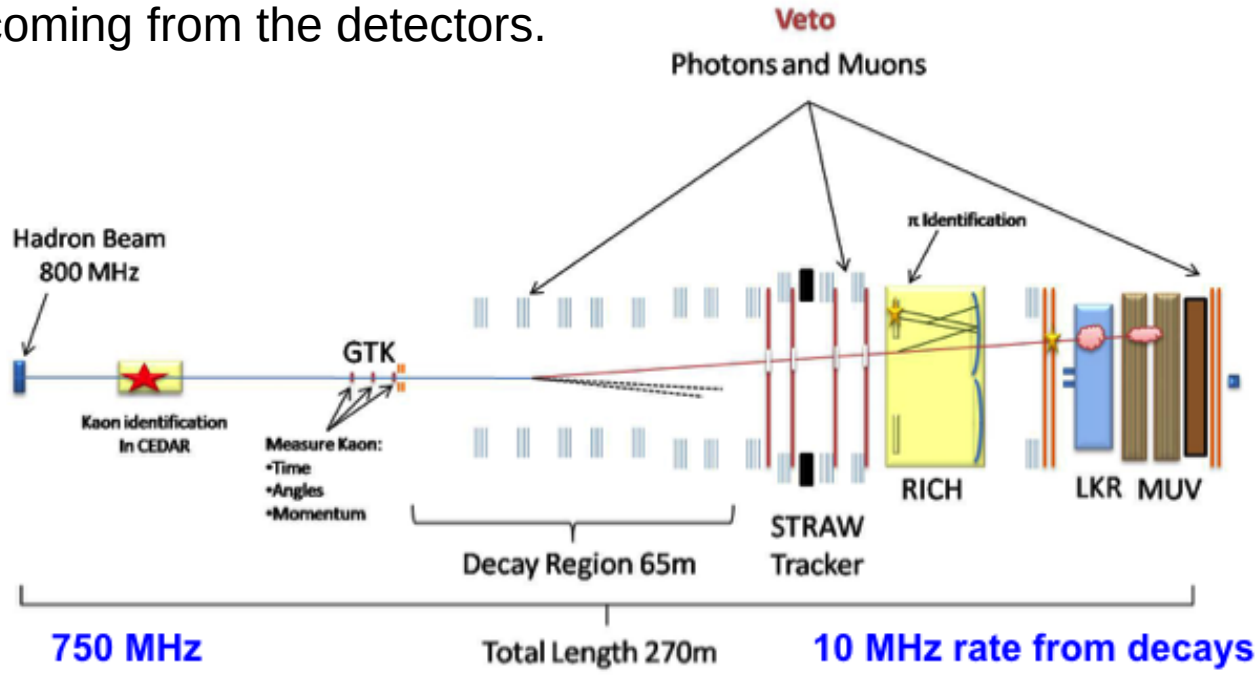
- Off-line (Monte Carlo simulations, analysis, …).
- Trigger-less readout systems
  - **Mu3e** (2 GHz event rate, multiple Tbit/s) - 2016
  - **PANDA** (20 MHz event rate, 200 GB/s) - 2018
- High Level Trigger
  - **Alice** (300 Hz event rate, 30 GB/s) - 2010
  - **LHC upgrade** (Atlas, CMS, LHCb) - 2018
- Low Level Trigger ← Our focus
  - **NA62**: pilot project investigating GPU usage in first trigger level (started 2011).
  - Hard real-time (10 MHz event rate, < 10 Gb/s, 1 ms).

GPU Real-time Computing (in HEP)

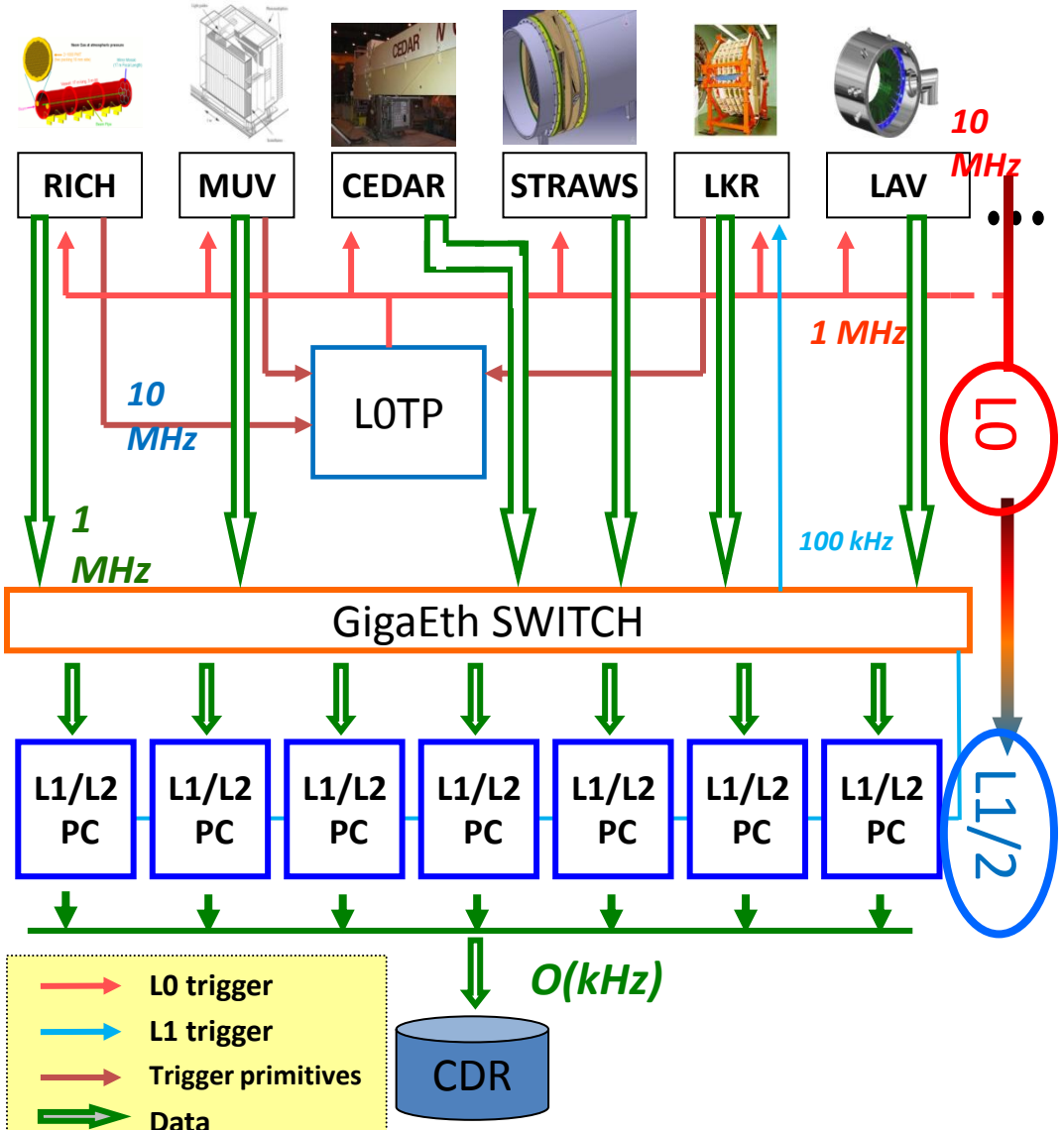
- **Hard real-time**: *a system that has to respond strictly within a given time budget to avoid failures (or data loss).*
  - Real time control systems (e.g. Tokamak plasma control)
  - First level (or low level) triggers
- **System latency**:
  - Is GPU processing latency small and stable enough for the given task?
  - Is the latency of network communications to and from GPU memory small and stable enough?
- **System throughput**:
  - Has the GPU enough computing power to execute the assigned task within the given time budget?

A Physics Case: NA62 Experiment at CERN

- Goal is the precision measurement of the BR of ultra-rare decay process
  - $K^+ \to \pi^+ \nu \bar{\nu}$
  - High precision theoretical prediction: sensitive to new physics
- Many (uninteresting) events: 10^7 decays/s.
- Ultra-rare: 1 target event on 10^{10} particle decays
  - Expect to collect ~100 events in 2/3 years of operation
  - Need to select the interesting events, filtering efficiently and quickly the data stream coming from the detectors.

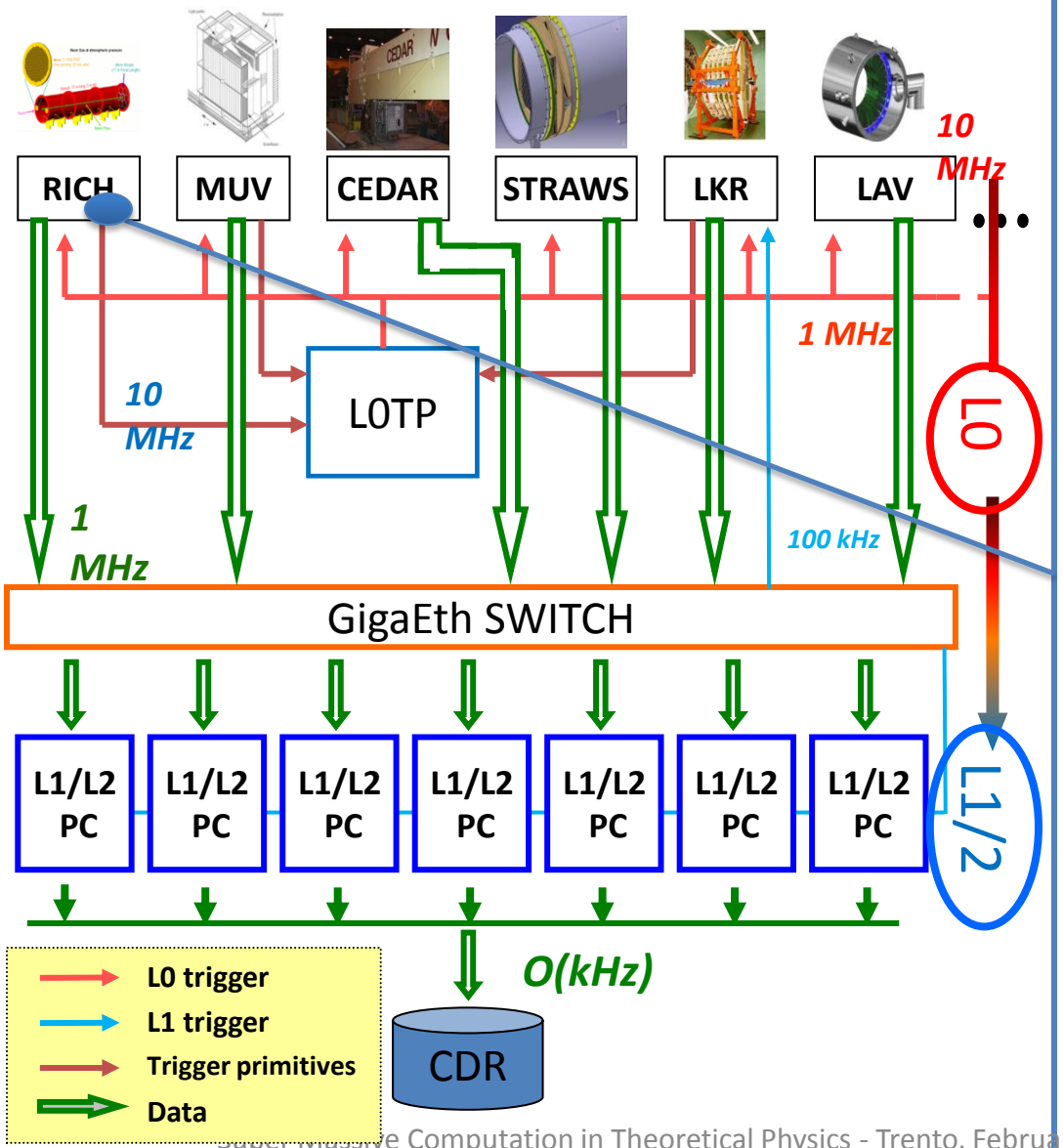


NA62 Trigger & DAQ



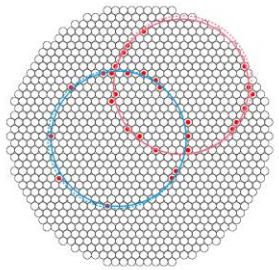
- **L0**: Hardware synchronous level
  - 10 MHz to 1 MHz, **1 ms max. latency**
  - Primitives (MUV, RICH, LAV, LKR)
- **L1**: Software level
  - "Single detector", 1 MHz to 100 kHz
- **L2**: Software level
  - "Complete information", 100 kHz to 10 kHz

Using GPUs in the NA62 L0 Trigger for the RICH



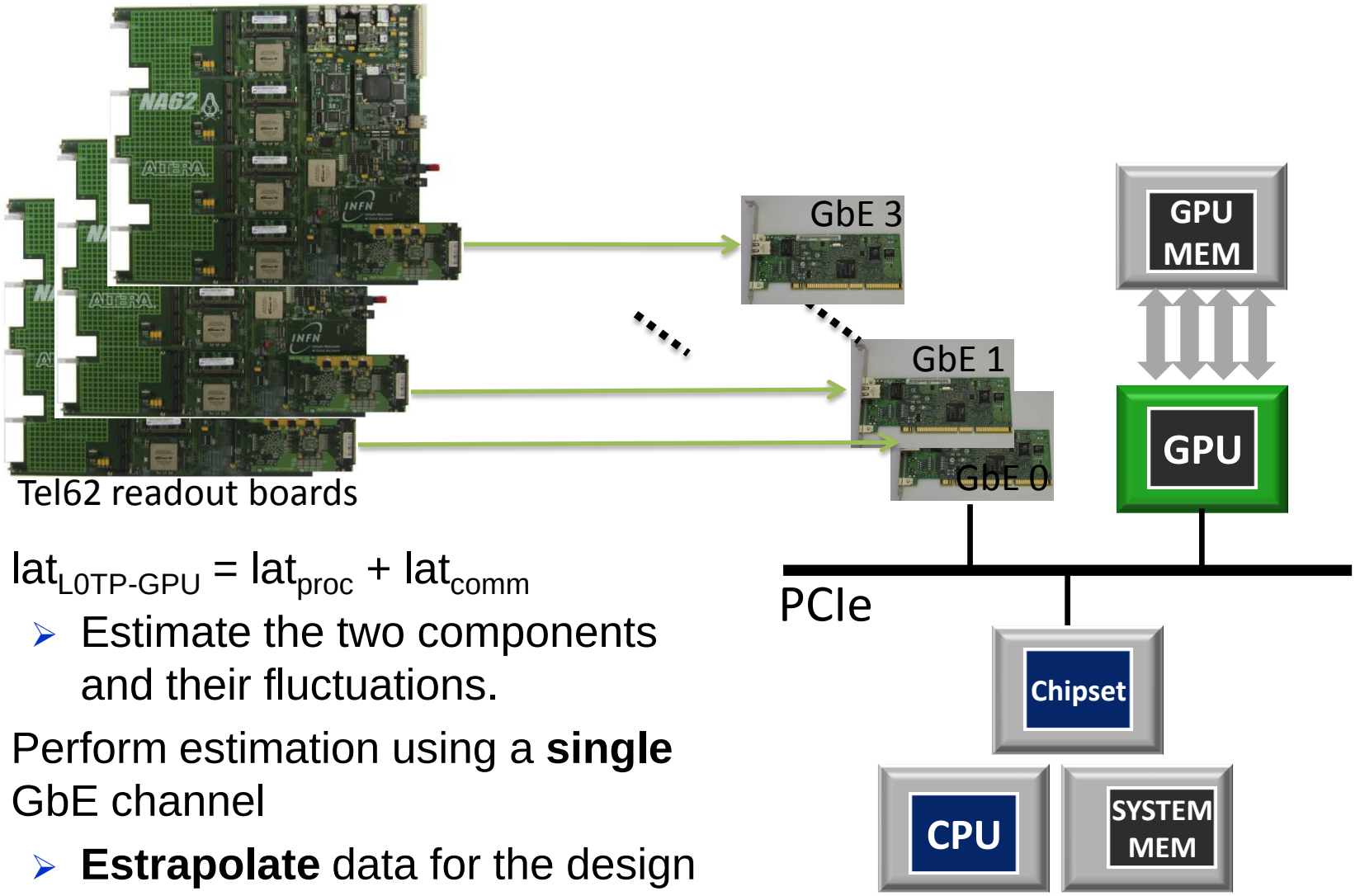
Replace custom electronics with a GPU-based system

- More selective trigger algorithms.
- Programmable.
- Upgradable.
- Rough detection of particle speed (radius) and direction (centre)



- **Efficient match of circular hit patterns** on GPUs.

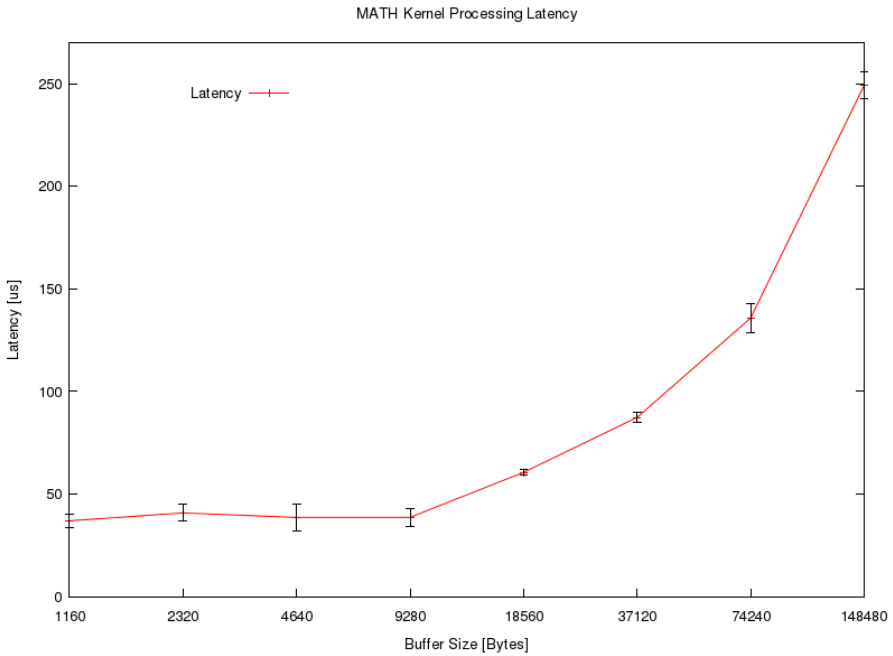
System Latency Estimation

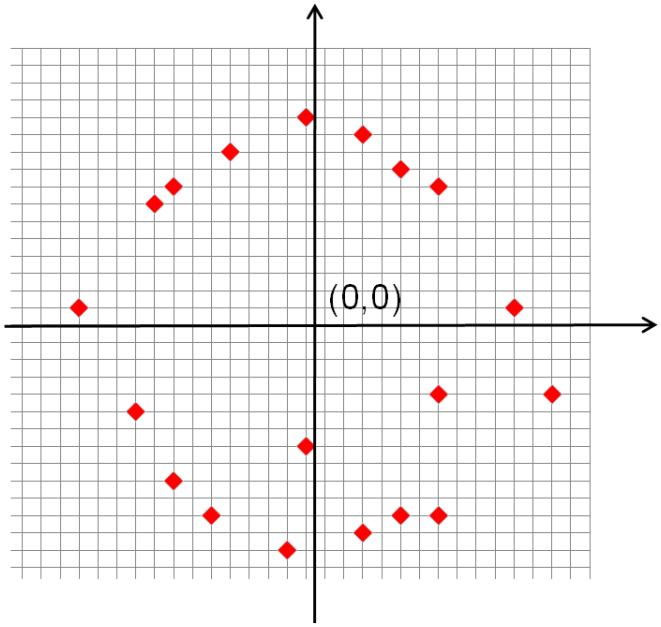


- $lat_{L0TP\text{-}GPU} = lat_{proc} + lat_{comm}$
  - Estimate the two components and their fluctuations.
- Perform estimation using a **single** GbE channel
  - **Estrapolate** data for the design of the "full bandwidth" system.

Processing Latency Estimation

- **lat_{proc}** : time needed to perform rings pattern-matching on the GPU with input and output data on device memory.



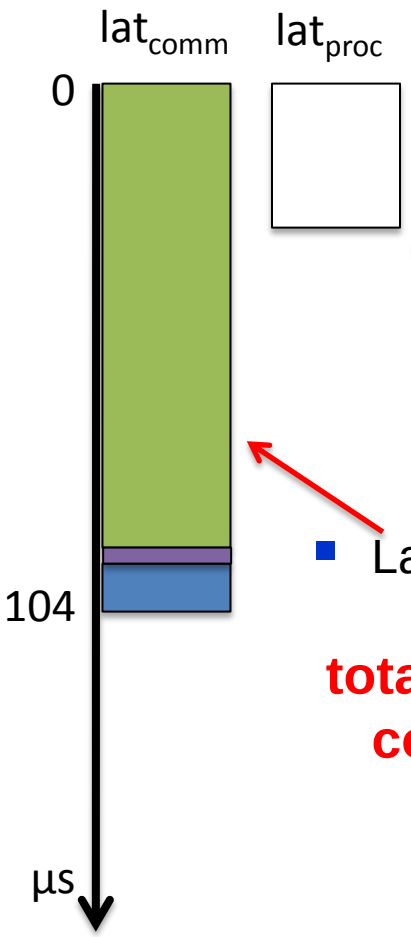


- **MATH** algo:. translation of the ring to centroid. In this system a least square method can be used. The circle condition can be reduced to a linear system, analitically solvable, without any iterative procedure.

Communication Latency
Standard GbE NIC / SW Stack

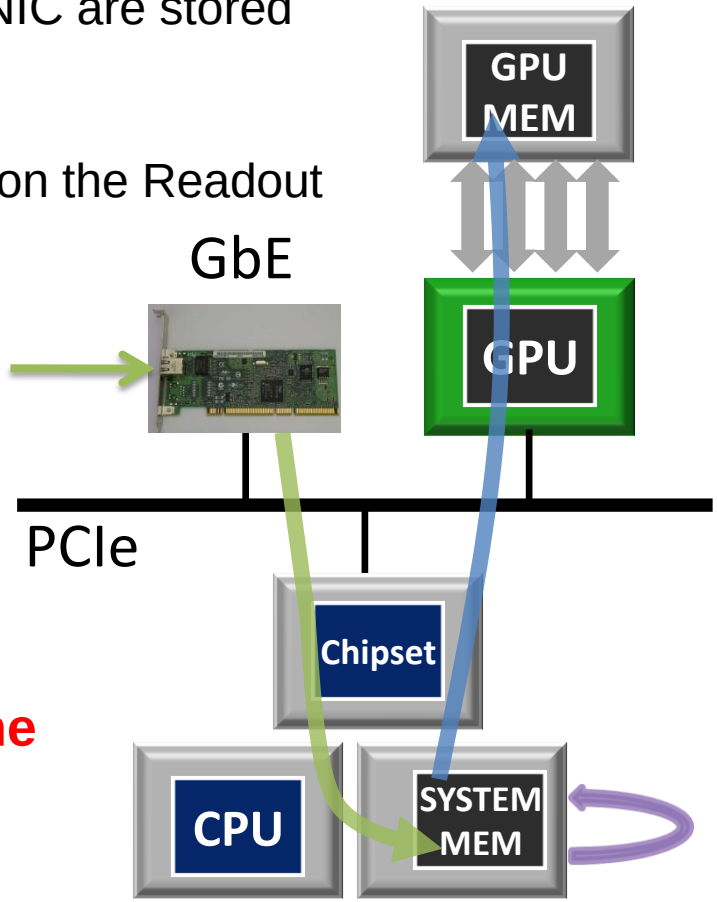
lat_{comm} : time needed to receive event data from GbE NIC to GPU memory.

- 40 events data (1400 bytes) sent from Readout board to the GbE NIC are stored in a receiving GPU buffer.
- T=0 start of send operation on the Readout Board.

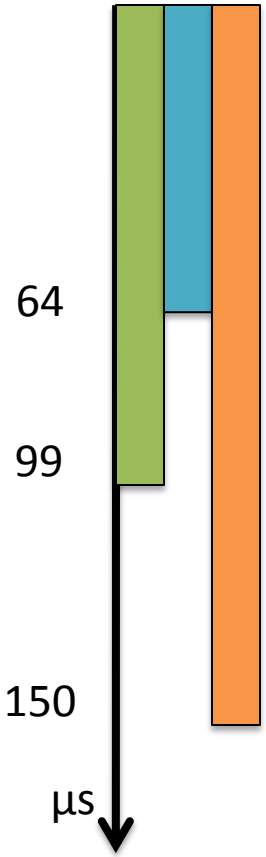


- $Lat_{comm} = 104\ us \cong 3.5 \times Lat_{proc}$

total latency is dominated by the communication component.

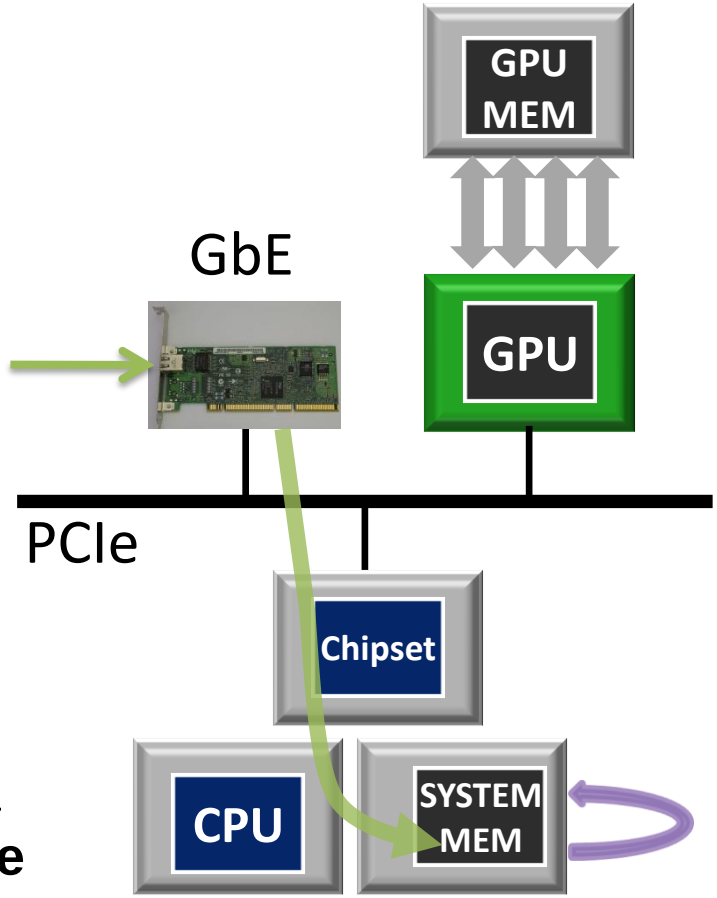


Communication Latency
Standard GbE NIC / SW Stack




```
sockperf: Summary: Latency is 99.129 usec
sockperf: Total 100816 observations; each
percentile contains 1008.16 observations

sockperf: ---> <MAX> observation = 657.743
sockperf: ---> percentile 99.99 = 474.758
sockperf: ---> percentile 99.90 = 201.321
sockperf: ---> percentile 99.50 = 163.819
sockperf: ---> percentile 99.00 = 149.694
sockperf: ---> percentile 95.00 = 116.730
sockperf: ---> percentile 90.00 = 105.027
sockperf: ---> percentile 75.00 = 97.578
sockperf: ---> percentile 50.00 = 96.023
sockperf: ---> percentile 25.00 = 95.775
sockperf: ---> <MIN> observation = 64.141
```



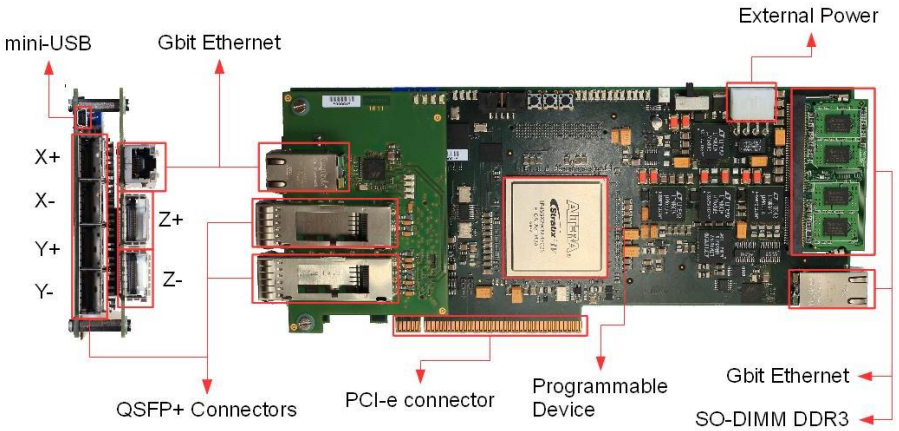
Fluctuations on latency of the data communication task **may hinder the real-time constraints** of the system.

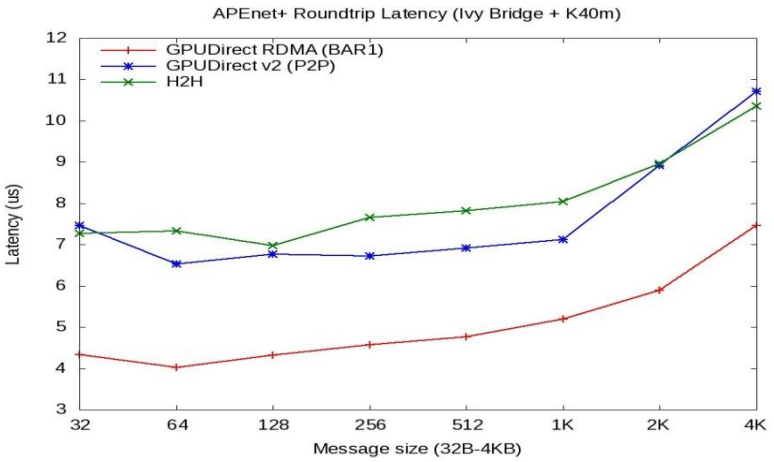
NaNet Design

- Challenge:
  - Lower latency of data communication task (and its fluctuations).
- How?
  1. Injecting directly data from the NIC into the GPU memory **without intermediate buffering.**
  2. **Offloading** the CPU from network stack protocol management, eliminating possible OS jitter effects.
- **NaNet design**:
  - a family of network cards dedicated to real-time systems (for HEP).
  - re-use part of the **APEnet+** design, implementing a **PCIe** interface with **GPUDirect P2P/RDMA** capabilities to get:
    - High bandwidth and low and stable communication latency.
    - Multiple link technologies and network protocols.
    - Processing on data streams.
    - Flexible, extensible, easily upgradable system
    - Optimized communication with GPU accelerators

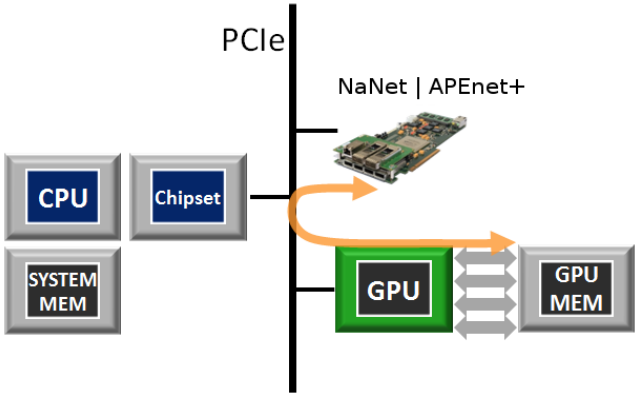
NaNet leverages on APEnet+….

See R.Ammendola talk!!!

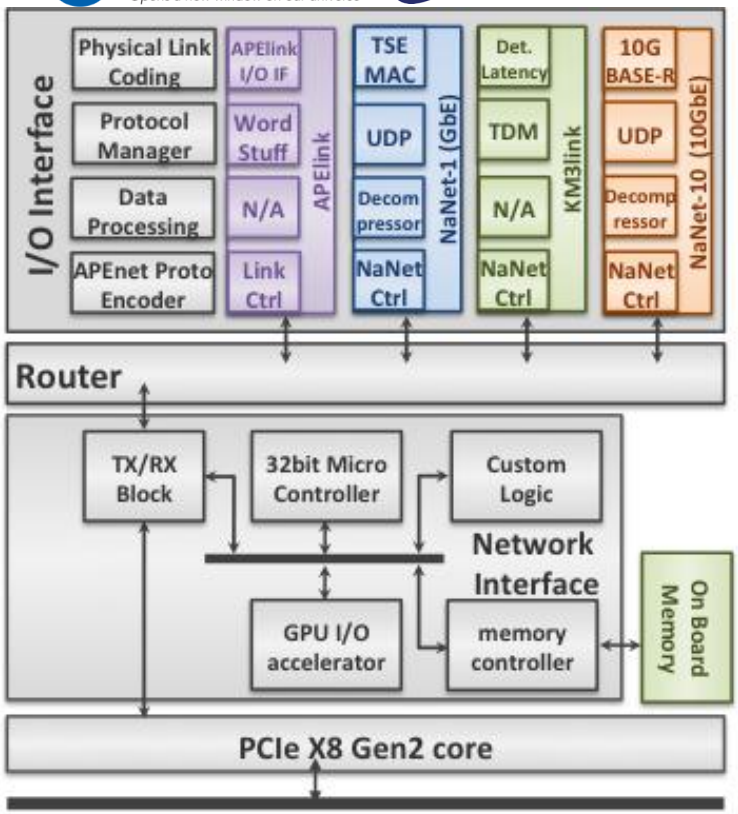


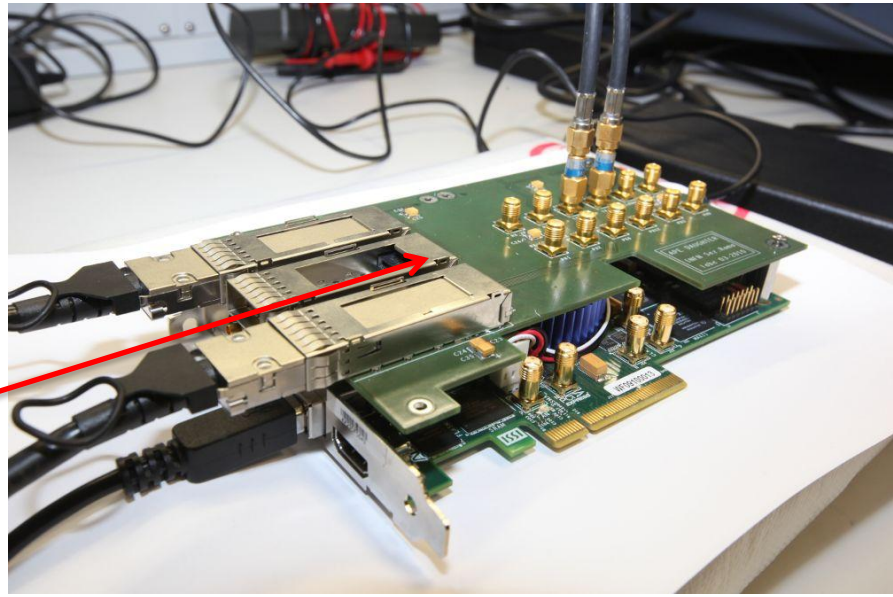


- High Performance Computing
- Point-to-point, low-latency, high-throughput NIC implementing a 2D/3D torus topology
- PCI Express x8 gen2
- Up to 6 fully bidir 34 Gbps/channel over QSFP+.
- RDMA Hardware support for CPU offload
- NVIDIA GPUDirect RDMA/P2P HW support
  - GPUDirect allows direct data exchange on the PCIe bus with no CPU involvement.
  - No bounce buffers on host memory .
  - **Latency reduction for small messages!!!**



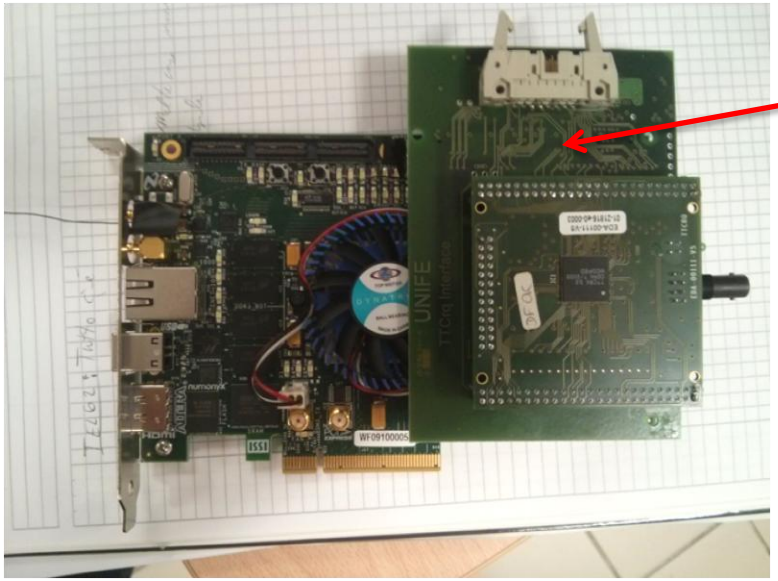
NaNet-1 Implementation





- Modular system (multiple physical)
- Implemented on Altera Stratix IV dev board (EP4SGX230KF40C2)
- Supports additional 3 APElink channels (20 Gb/s each) with HSMC daughtercard

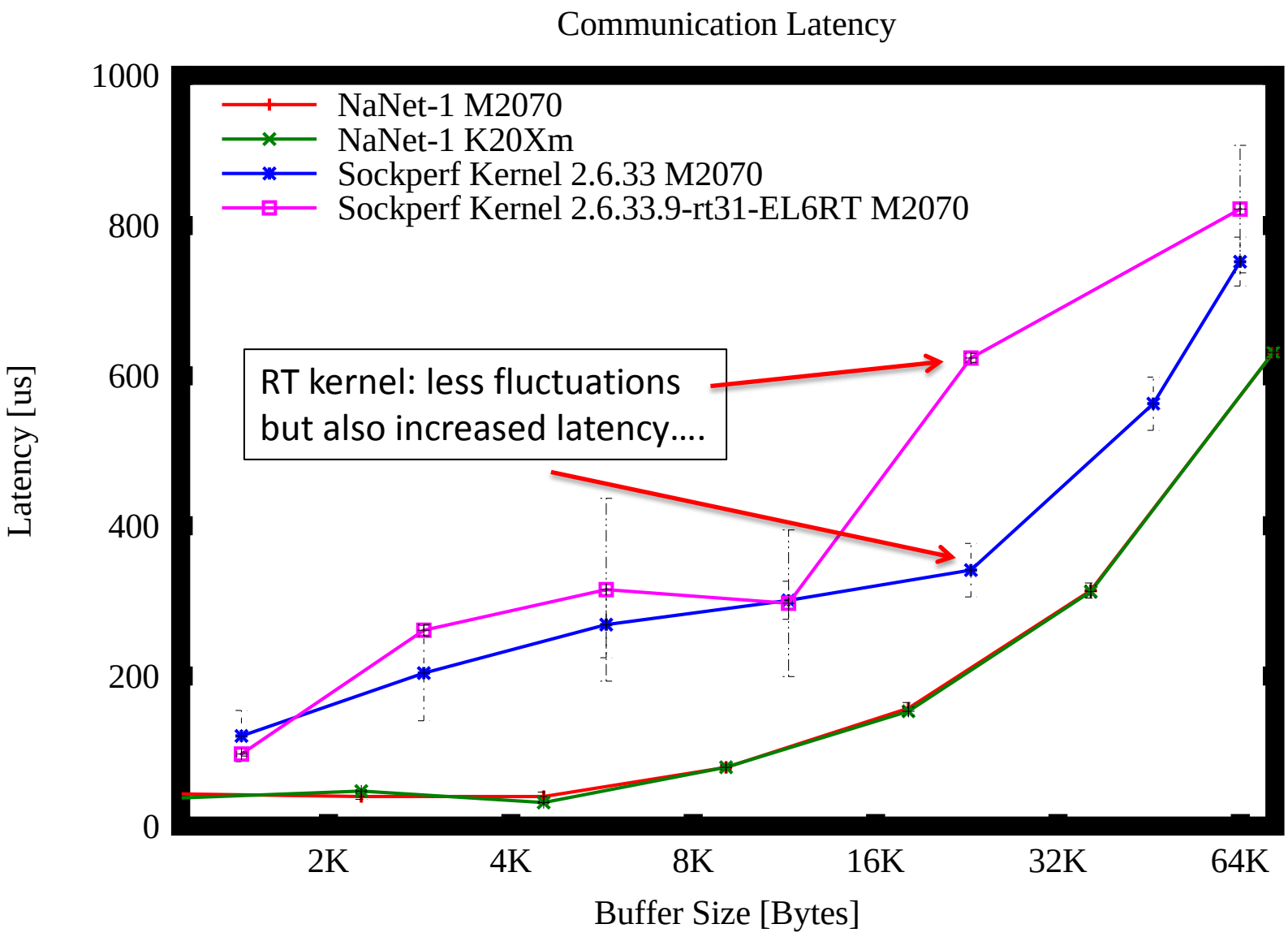
NaNet-1 Integration in NA62 DAQ & Trigger System



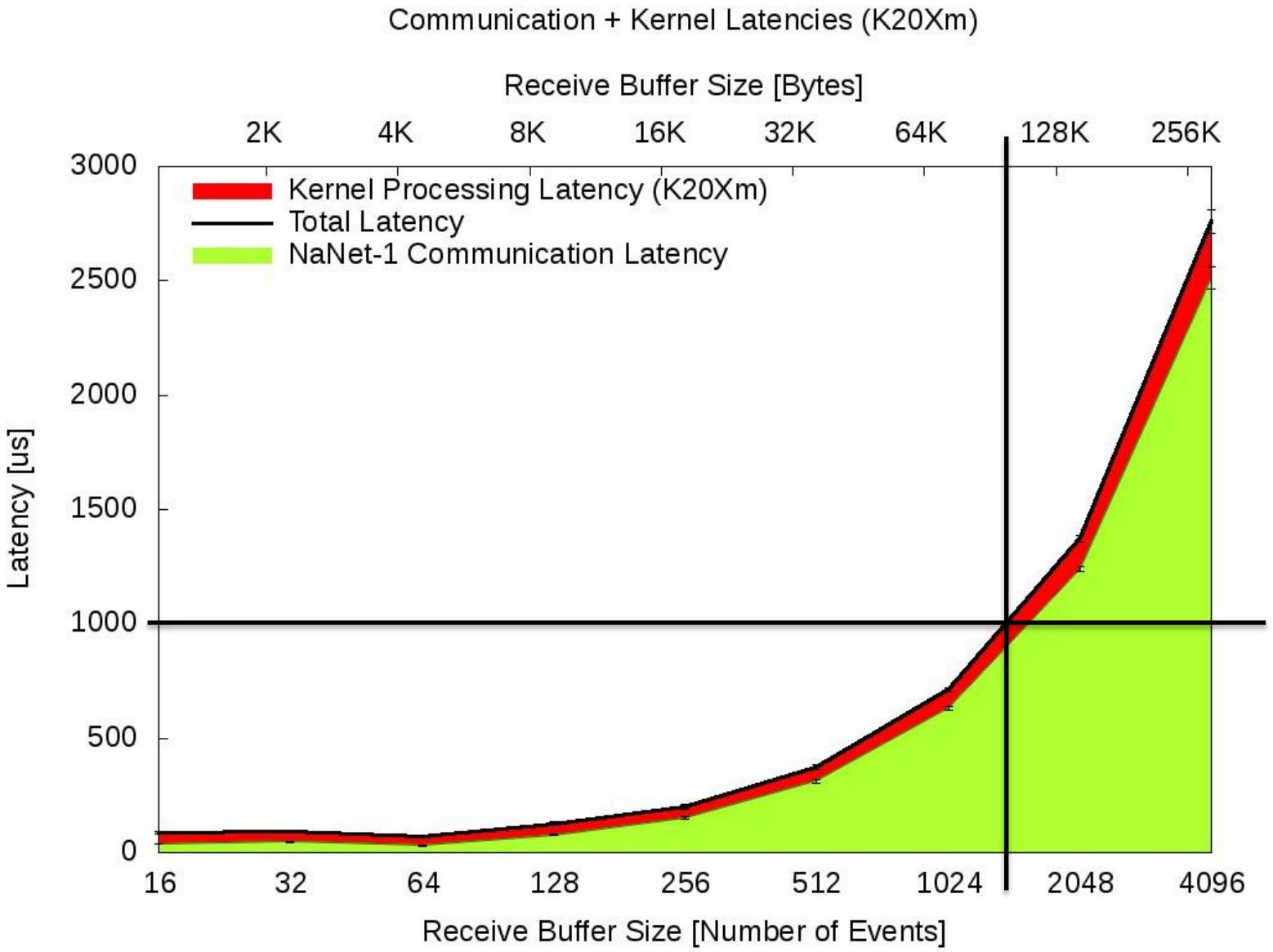
- TTC daughtercard with HSMC connector (INFN Ferrara)
- NaNet receives TTC stream with timing (40 MHz clock, SOB, EOB) and trigger signals from the experiment.
- Allows synchronous operation (accurate latency measurements)

- First integration and tests activities during August technical run.
- Parasitic operation of GPU-based L0 trigger using NaNet-1 demonstrated end 2014.

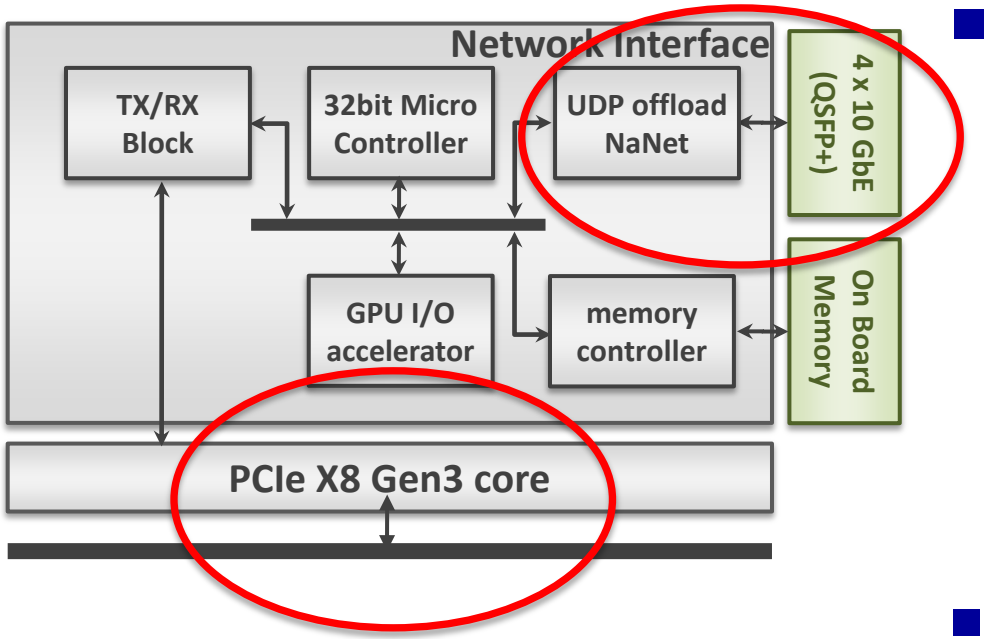
NaNet-1 Communication Latency



GPU-Based RICH L0-TP using NaNet-1 (Kepler K20X)

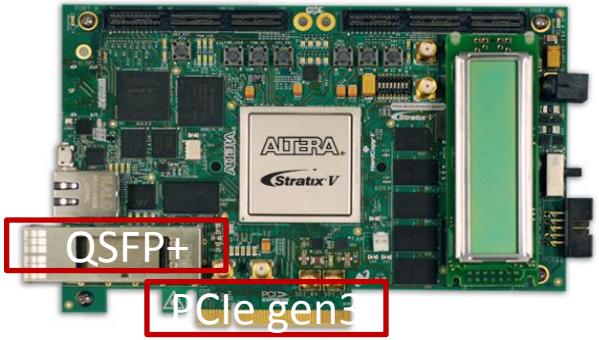


NaNet-10 and last generation FPGAs (Stratix V)

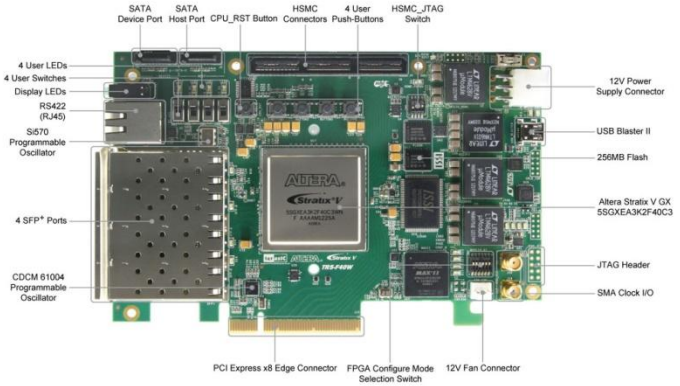


- Will be implemented on the Altera Stratix V dev board but porting on cheaper board (Terasic TR5-F40W) is possible
  - Four 10 GbE SFP+ ports
  - PCIe Gen3 (8 GB/s)
  - Faster embedded Altera transceivers (up to 14.1 Gbps)
  - hardened 10GBASE-R PCS
- Ready for Spring 2015 run

QSFP+ to 4 SFP+ cable



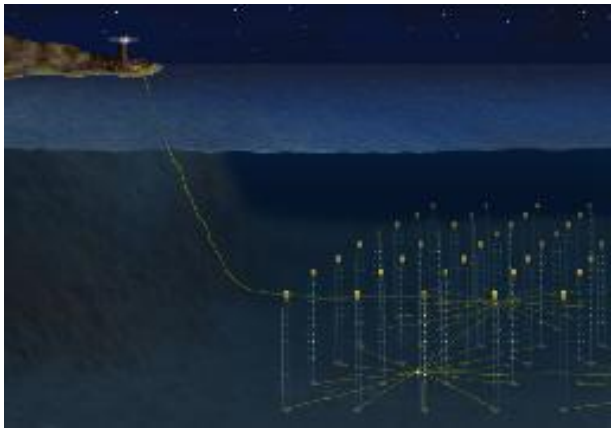
Altera Stratix V dev board

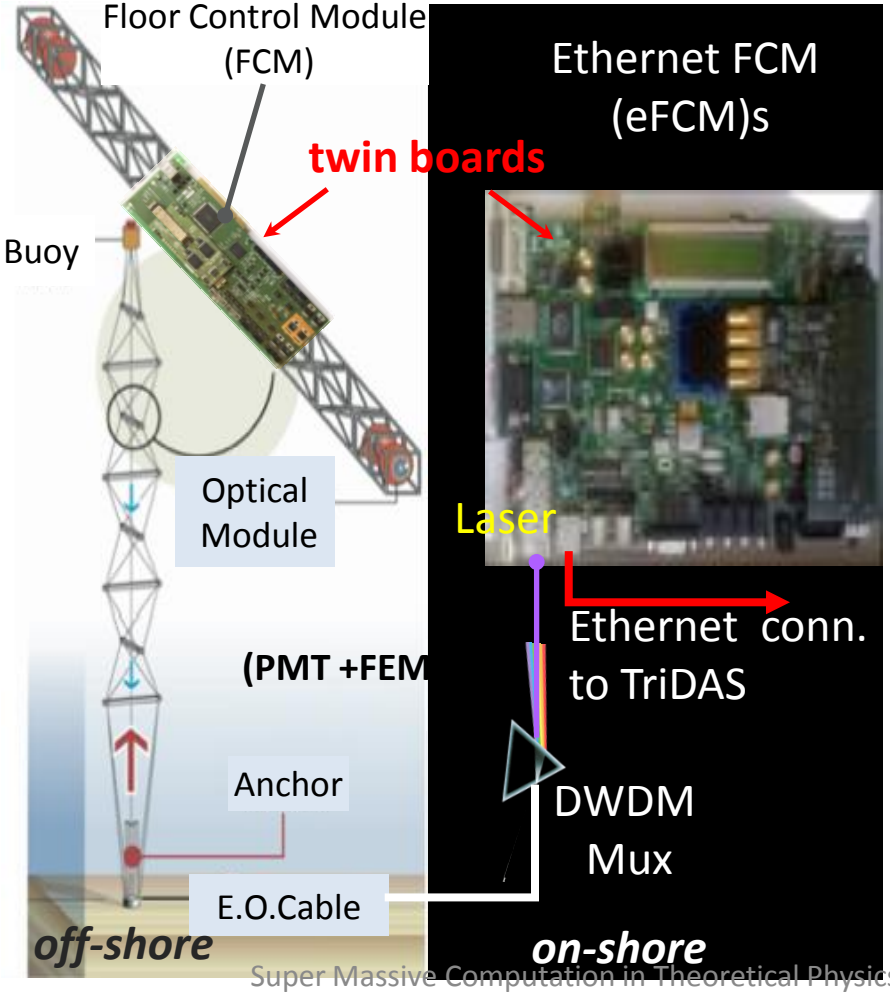


Terasic TR5-F40W

KM3Net-IT Experiment

- **KM3Net-IT**: an European deep-sea research infrastructure, hosting a neutrino telescope with a volume of cubic kilometer at the bottom of the Mediterranean Sea.

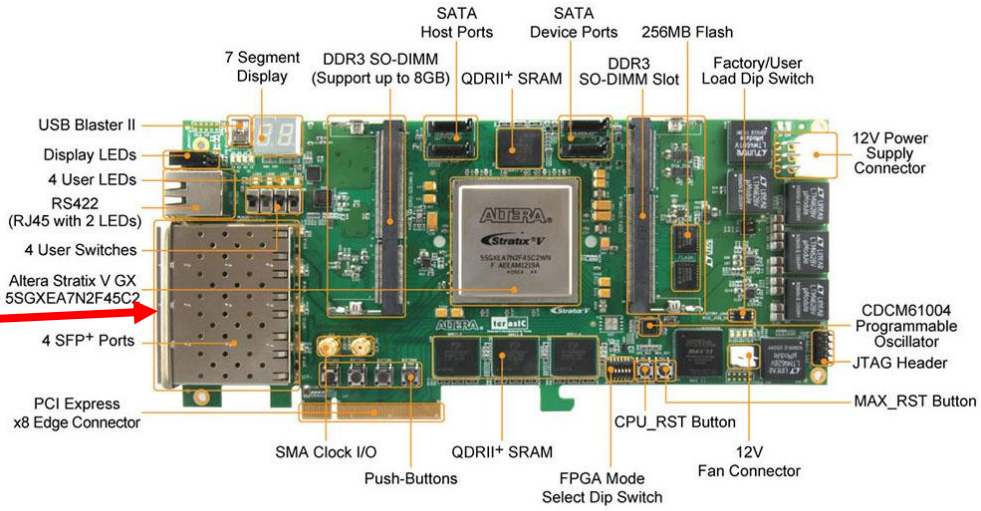




- Current read-out system design employs a large number of NO state-of-the-art components
  - 2.5 Gb/s optical link per 800Mb/s payload
  - 2 twinned (on and off-shore) FCM boards per floor (14 floors per tower)
  - Many servers for HW read-out hosting
- When scaling to Km^3 many cost/size/power/reliability issues.

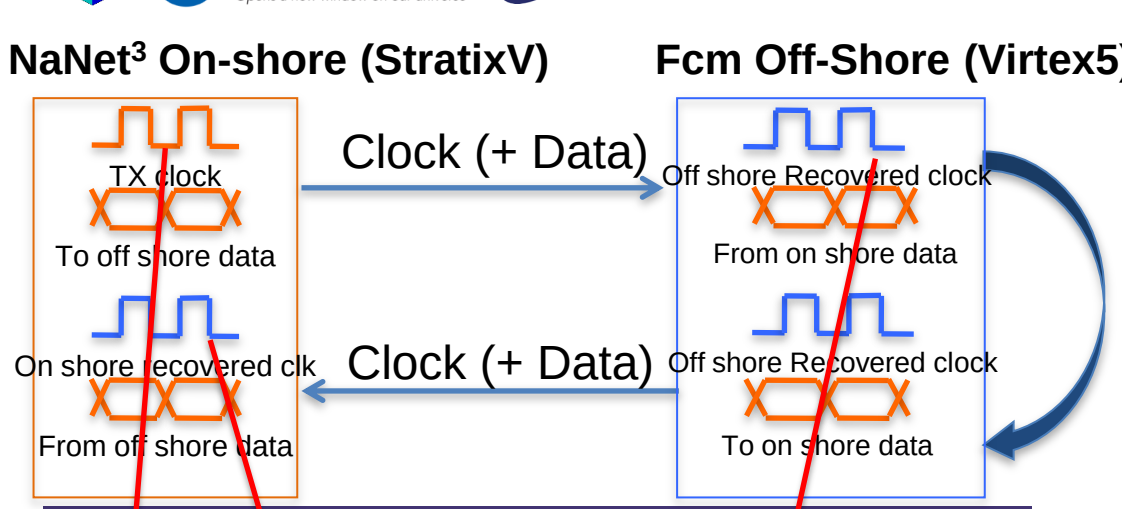
Why NaNet³?

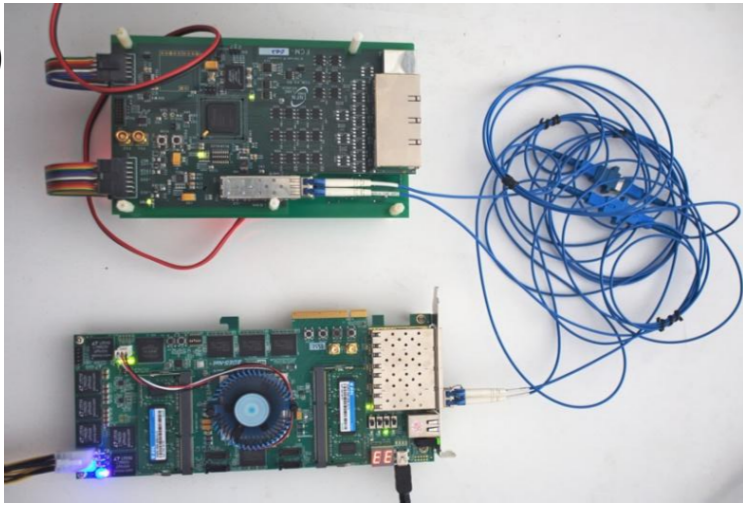
- Device: TERASIC DE5-NET
- Based on Altera Stratix V FPGA
- PCIe Gen2 x8 (Gen3 support)
- 4 independent SFP+ ports (up to 10Gb/s)

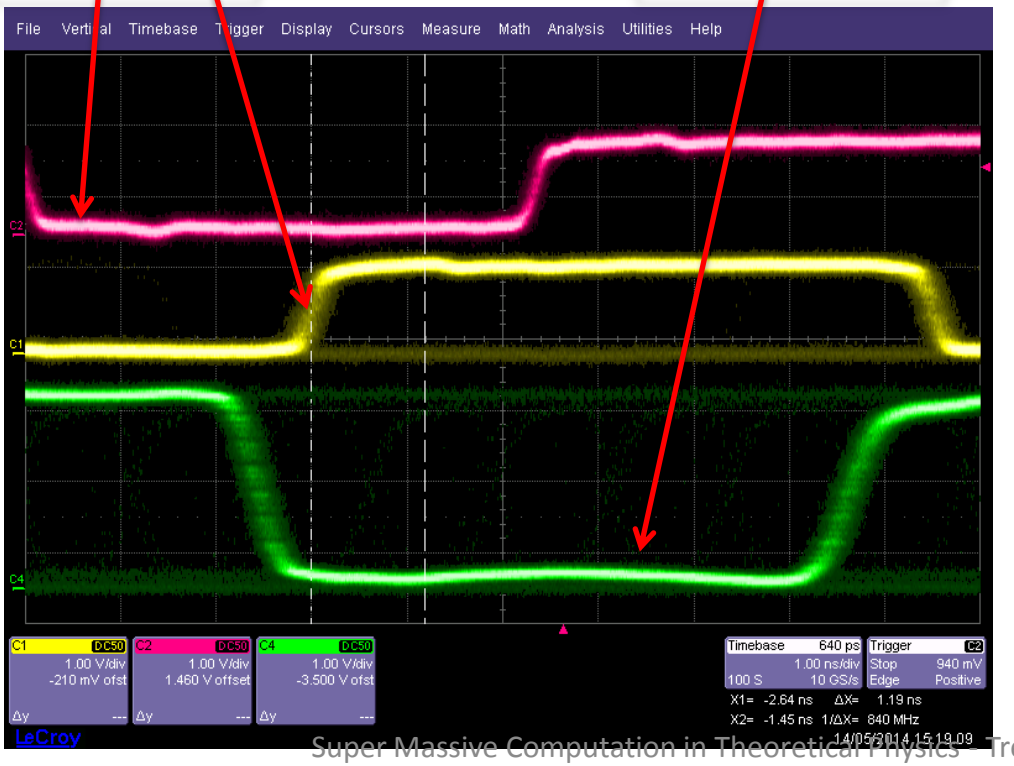


- NaNet³ specifications for enhanced KM3Net-IT read-out system:
  - Deterministic latency link :"Fixed Latency" clock distribution for under-water events time-stamping
  - Time Division Multiplexing protocol support
  - 4 channels/PCIe board (i.e. less servers, less read-out boards,…)
- Further possible enhancements (if required by final setup specifications):
  - GPUDirect P2P/RDMA capability: to support GPU-based trigger developments
  - Link speed up to 10Gb/s: Further reduction of a factor 4 in number of optical channels is possible
  - PCIe Gen3 implementation

Deterministic latency link inter-operability



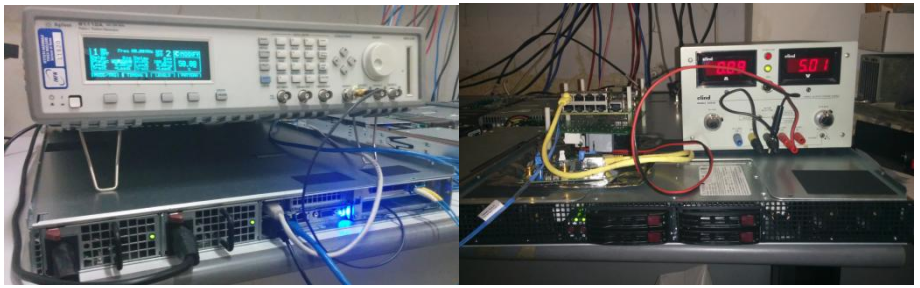




- Testbed: FCM vs Terasic DE5-Net
  - Custom hw mode for FCM Transceivers (Xilinx)
  - Latency deterministic mode for Stratix V Transceiver
  - 2mt copper and 2 mt long fiber
- Test:
  - 12 hours of periodic (~s) Tx clock reset to verify pll locking and rx word alignment

NaNet³ current status

- Testbed Environment
  - FCMserver
  - NaNet³
  - FCM + FEM (off-shore systems)
- Echo test (single channel)
  - NaNet/FCM roundtrip
- Slow-control test
  - TX: from NaNet³ to FCM
- Data Acquisition test
  - Frame ID in slow_control_0
  - Data integrity
  - 72 hours test passed
- Work-in-progress
  - Integration with FCMserver application (also to validate single channel implementation)
  - Multi-channel support (hardware ready, test in progress…)
- Deployment on final site starting from Feb 2015



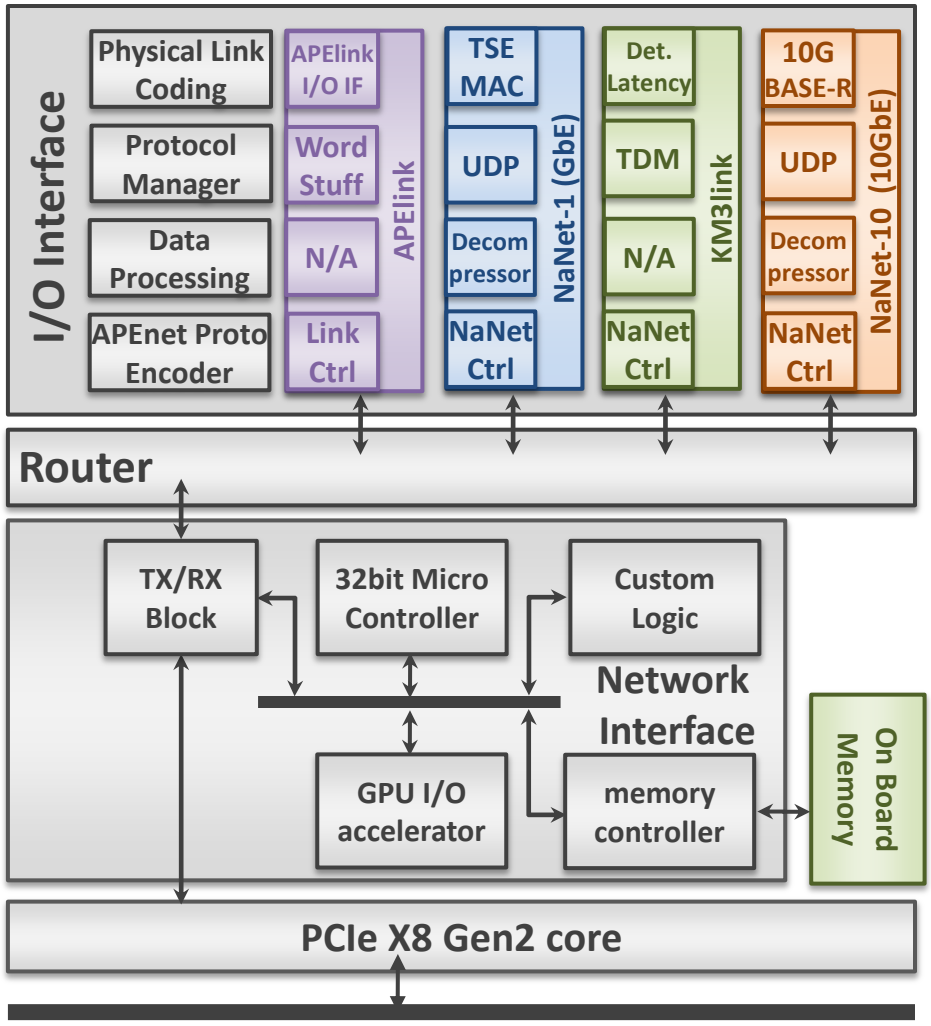
Conclusions

- NaNet design has proved to be effective in two different experimental contexts thanks to its modularity and high performance/low latency (RDMA GPUDirect, network protocol offloading) features.
  - NA62 GPU-based Low Level Trigger
    - Demonstrated real-time data communication between the NA62 RICH read-out system and the GPU-based L0 trigger processor over a single GbE link.
    - Demonstrated scalability up to multiple 10GbE read-out channels.
    - 4 ports 10GbE version ready soon (2Q2015).
  - KM3Net-IT on-shore read-out system:
    - Demonstrated different vendors, high-end FPGAs serial links inter-operability (with Deterministic Latency).
    - Compliant with Real-time constraints of TDM processing.
    - Ready for future enhancements (on-shore link aggregation, GPU-based trigger).
  - More to come….

- Leverage on SUMA technological R&D activities we created convergence between HPC systems and HEP experiments low-level trigger computing

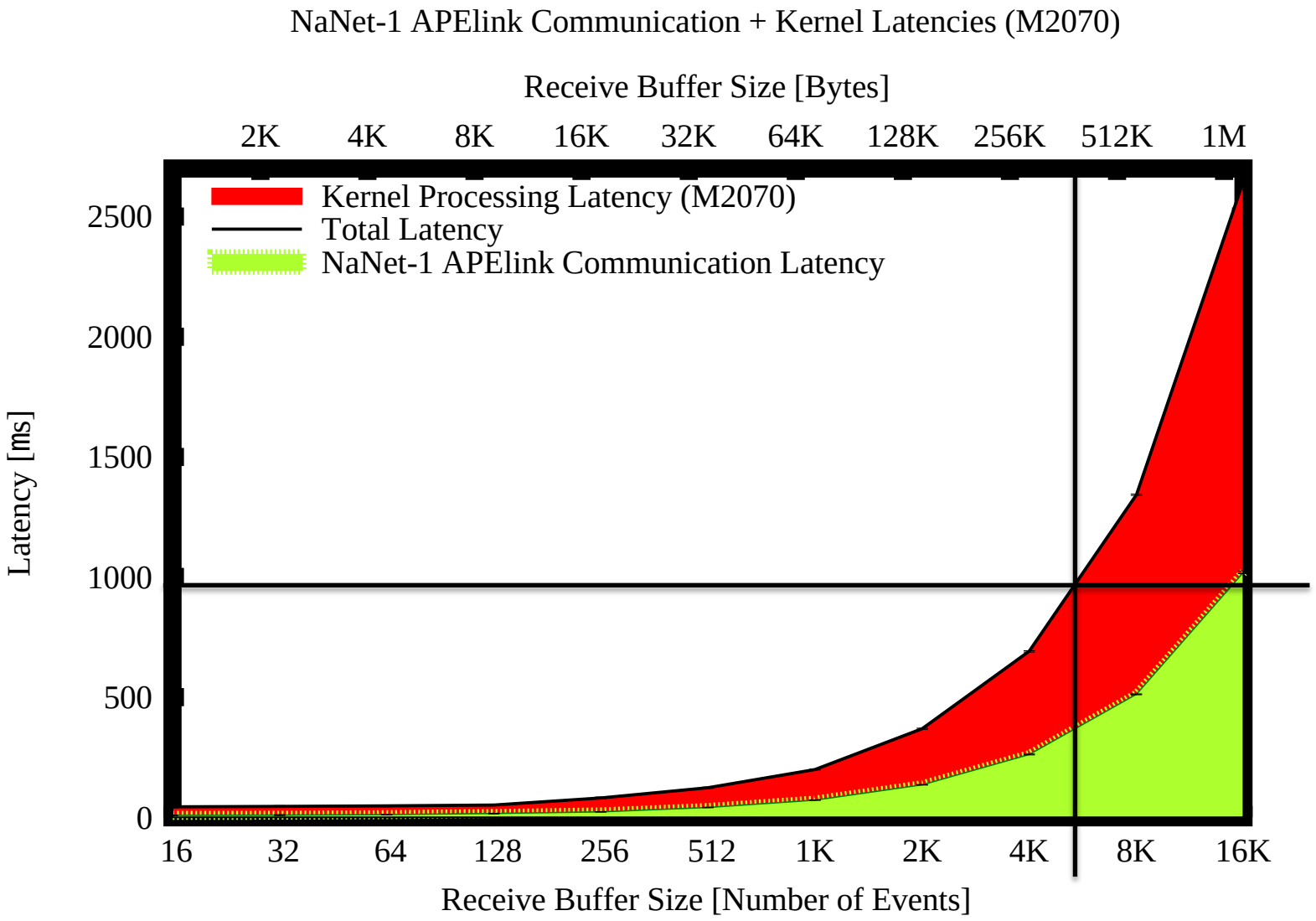
Backup

NaNet Modular Design

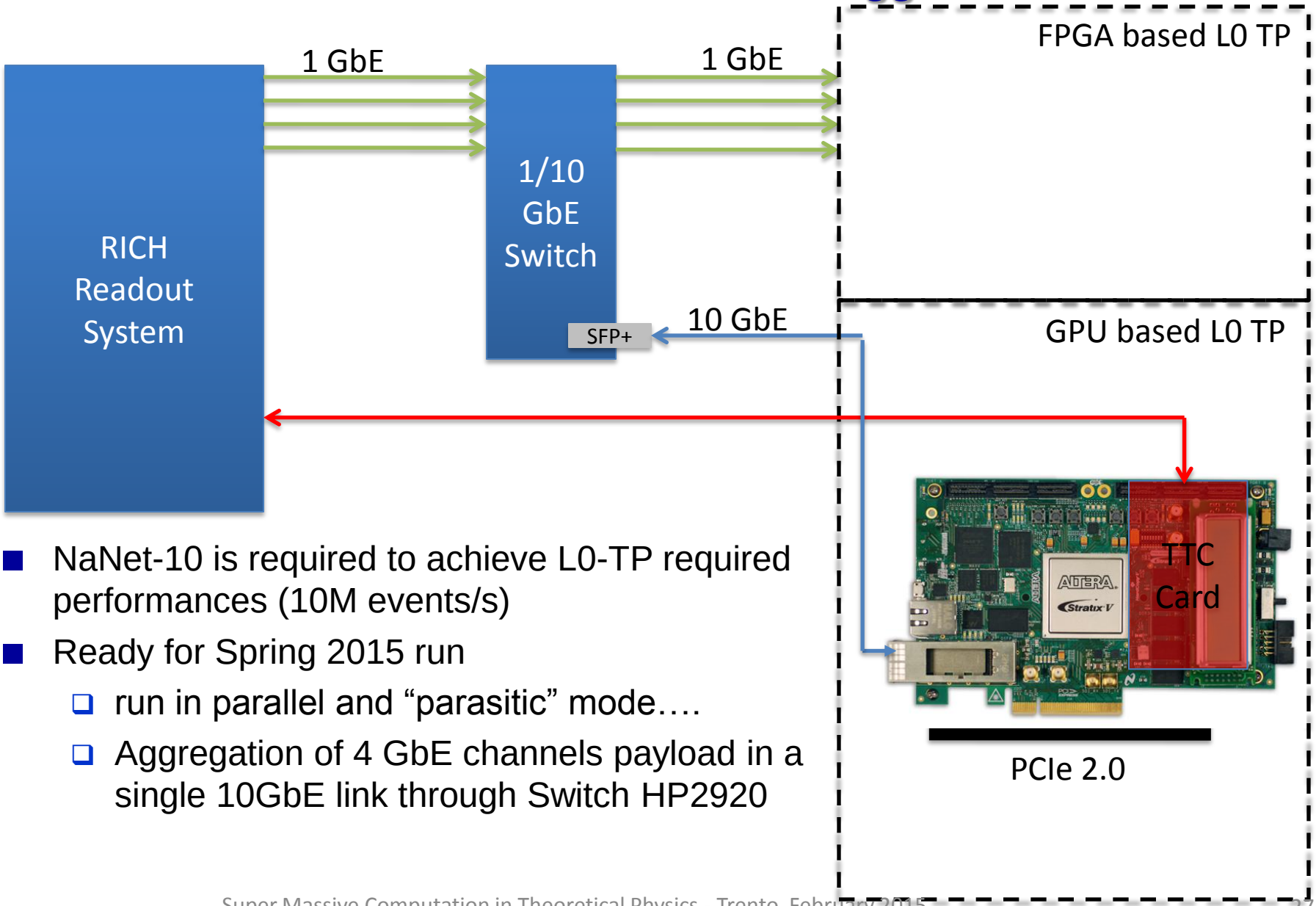


- I/O Interface
  - Multiple link.
  - Multiple network protocols.
- Router
  - Dynamically interconnects I/O and NI ports.
- Network Interface
  - Manages packets TX/RX from and to CPU/GPU memory.
  - Microcontroller.
- PCIe X8 Gen2 Core
  - CPU BW: 2.8 GB/s Read, 2.5 GB/s Write.
  - GPU BW: 2.2 GB/s Read, 2.5 GB/s Write.

GPU-Based RICH L0-TP using NaNet-1 (APElink):



NaNet-10: path towards the final NA62 RICH L0 GPU Trigger Processor



- NaNet-10 is required to achieve L0-TP required performances (10M events/s)
- Ready for Spring 2015 run
  - run in parallel and “parasitic” mode….
  - Aggregation of 4 GbE channels payload in a single 10GbE link through Switch HP2920

Time Division Multiplexing (TDM) support

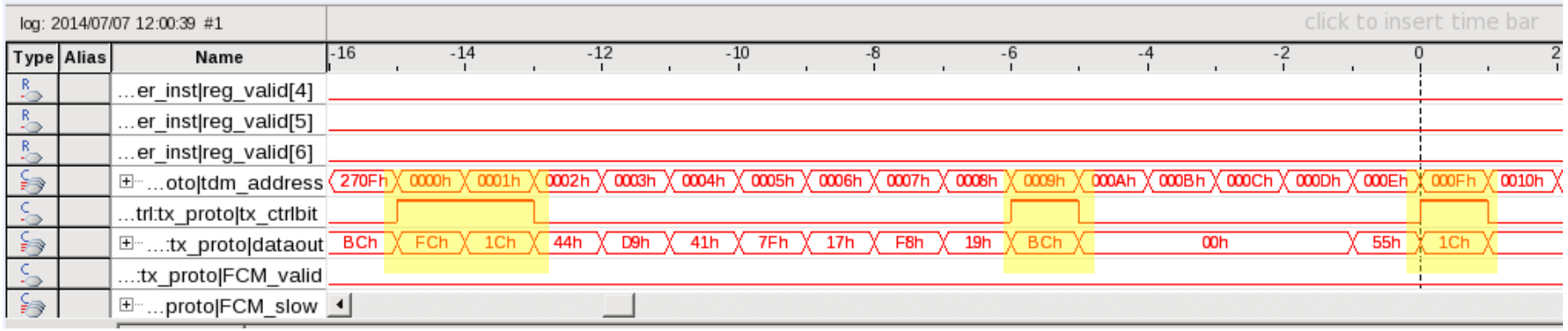
NaNet I/O interface Customization:

- TDM implementation: compliant with KM3Net-IT specification, echo test OK!

	K Code	Kout	D
IDLE_CODE	K28.5	1	0xBC
FRAME_CODE	K28.7	1	0xFC
DATA_CODE	K28.0	1	0x1C

Field	Validity Code	Start Address	Stop Address	Length [byte]
Start Frame	FRAME_CODE	0x0000	0x0000	1
Frame Time	DATA_CODE	0x0001	0x0008	1+6+1
FCM Slow Control	DATA_CODE	0x0009	0x000E	1+4+1
OM0 Slow Control	DATA_CODE	0x000F	0x0014	1+4+1
OM1 Slow Control	DATA_CODE	0x0015	0x001A	1+4+1
OM2 Slow Control	DATA_CODE	0x001B	0x0020	1+4+1
OM3 Slow Control	DATA_CODE	0x0021	0x0026	1+4+1
OM4 Slow Control	DATA_CODE	0x0027	0x002C	1+4+1
OM5 Slow Control	DATA_CODE	0x002D	0x0032	1+4+1
Hydro data	No IDLE_CODE	0x0200	0x03FF	512
OM0 Phy Data	No IDLE_CODE	0x0400	0x07FF	1024
OM1 Phy Data	No IDLE_CODE	0x0800	0x0BFF	1024
OM2 Phy Data	No IDLE_CODE	0x0C00	0x0FFF	1024
OM3 Phy Data	No IDLE_CODE	0x1000	0x13FF	1024
OM4 Phy Data	No IDLE_CODE	0x1400	0x17FF	1024
OM5 Phy Data	No IDLE_CODE	0x1800	0x1BFF	1024

- NaNet^3 TX



- NaNet^3 RX:

  Data flow: NaNet^3 → FCM board → NaNet^3