APEnet+ 34 Gbps Data Transmission System and Custom Transmission Logic

Andrea Biagioni – INFN

andrea.biagioni@roma1.infn.it

TWEPP 2013, Perugia, 26/09/2013

Outline

- Presentation of the APE project
- APEnet+ overview
- Description of the data transmission system
- The transmission control logic
- Conclusions

Outline

- ❑ Presentation of the APE project
- ❑ APEnet+ overview
- ❑ Description of the data transmission system
- ❑ The transmission control logic
- ❑ Conclusions

The People Involved

Alessandro Lonardo

Pier Stanislao Paolucci

Davide Rossetti

Piero Vicini

Roberto Ammendola

Andrea Biagioni

Ottorino Frezza

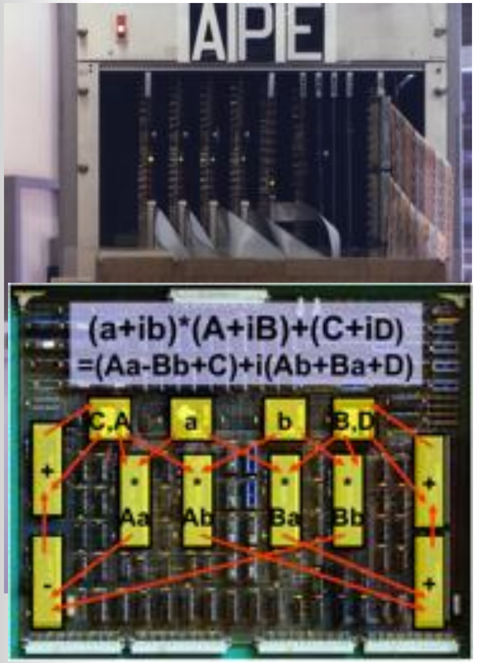
Francesca Lo Cicero

Francesco Simula

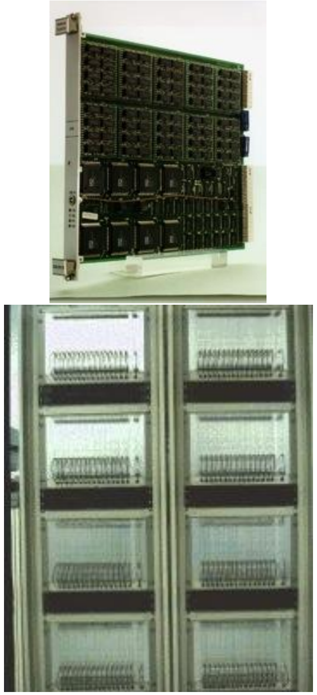
Laura Tosoratto

The APE History

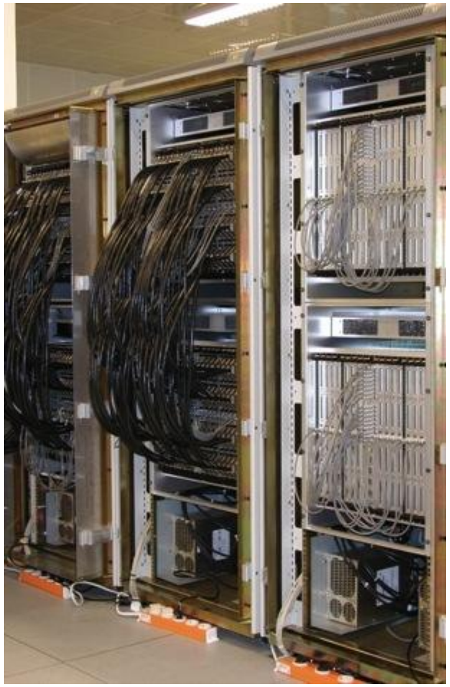
- APE (Array Processor Experiment) is a 25 years old project:
  - Custom HPC supercomputers: APE (86), APE100 (94), APEmille (99), apeNEXT (04)
  - MPP & PC Cluster interconnection network (apeNET)
  - FP Engine optimized for applications + dedicated 3D Torus interconnection network



APE1 – 1 gigaFLOPS



APE100 – 25 gigaFLOPS



APE1000 – 128 gigaFLOPS

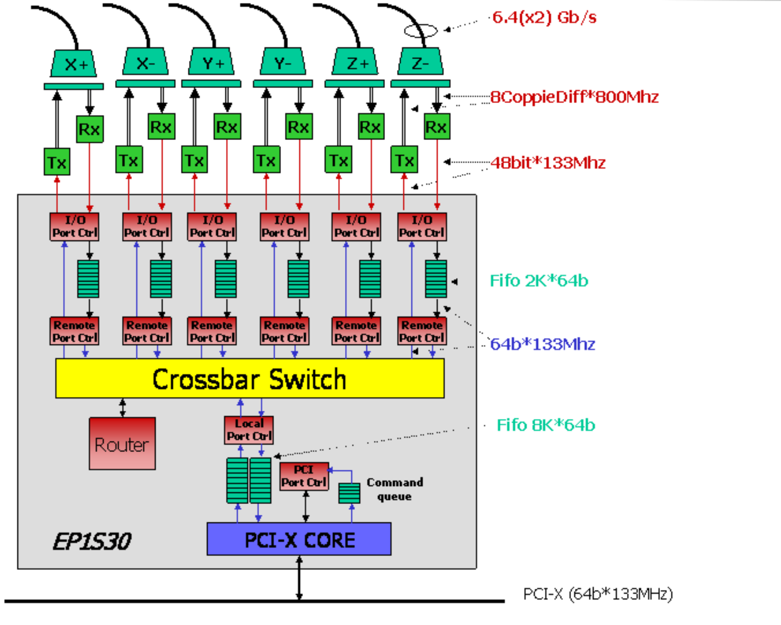
apeNEXT – 800 gigaFLOPS

APEnet: PC Cluster 3-d torus network

- ❑ Integrated routing and switching capabilities
- ❑ High throughput, low latency, “light-weight” protocol
- ❑ PCI Interface, 6 Links full-bidir on torus side

<u>APEnet history:</u>

- ❑ 2003-2004: APEnet V3 (PCI-X)
- ❑ 2005: APEnet V3+, same HW with RDMA API
- ❑ 2006-2009: APEnet goes embedded, integrated with ARM9
  - AMBA-AHB
  - DNP, Distributed Network Processor
  - EU SHAPES project co-development
- ❑ 2011: APEnet V4 aka APEnet+
  - PCIe gen2
  - NVIDIA GPU acceleration
  - EU EURETILE project co-development



Outline

- ❑ Presentation of the APE project
- ❑ APEnet+ overview
- ❑ Description of the data transmission system
- ❑ The transmission control logic
- ❑ Conclusions

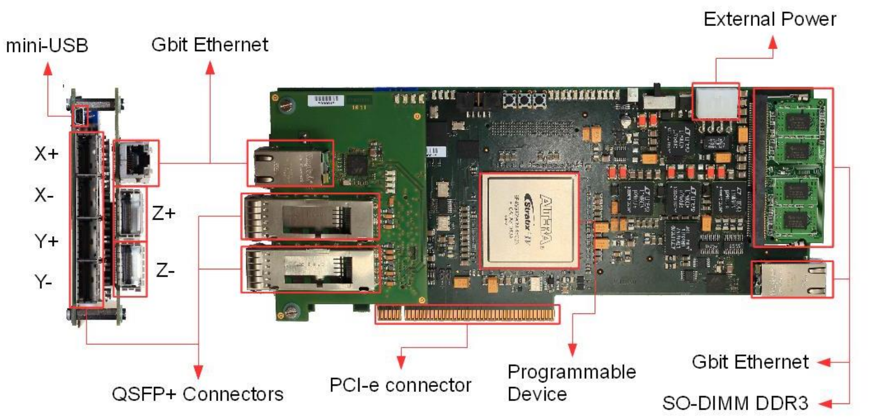
APEnet+

3D Torus Network:

- ❑ Scalable (today up to 32K nodes)
- ❑ Cost effective: no external switches

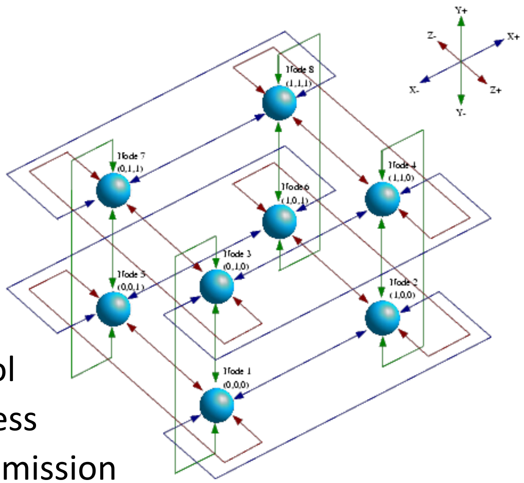
APEnet+ Card:

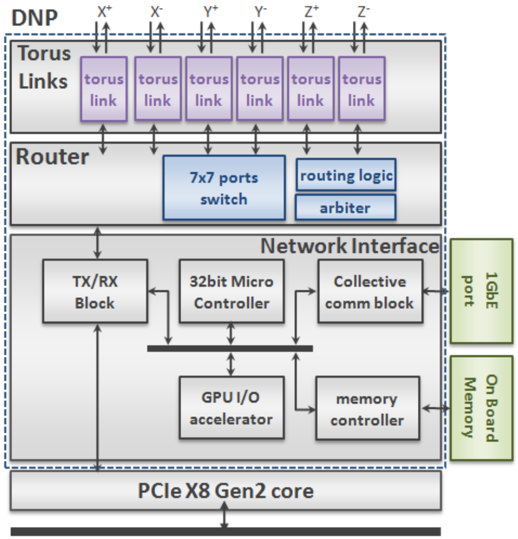
- ❑ FPGA based (ALTERA EP4SGX290)
- ❑ PCIe X8 Gen2 in X16 slot (peak BW 4+4 GB/s)
- ❑ 6 Full bidirectional torus links (68 Gbps)
  - ~ 400 Gbps aggregated raw bandwidth
- ❑ Industry standard QSFP+ cables (40 Gbps)



APEnet+ based on DNP:

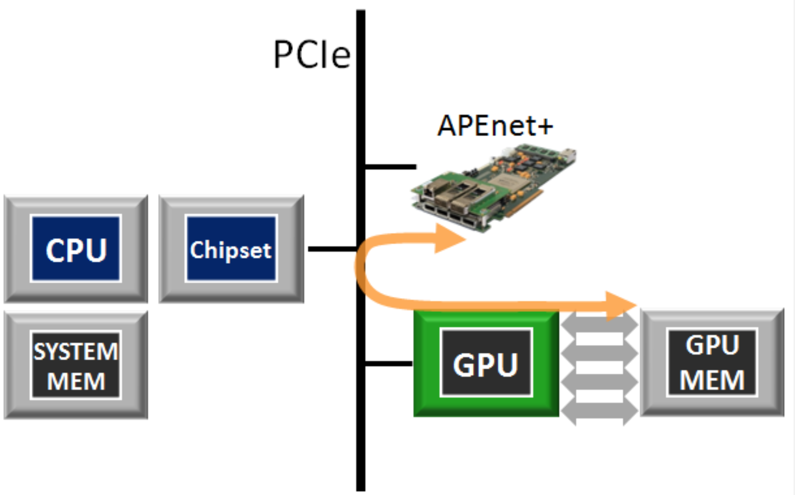
- ❑ Network Interface
  - RDMA: Zero-copy RX & TX!
    - o RDMA support for GPUs!
    - o Small latency and high bandwidth
  - Direct GPU interface, NVIDIA P2P
- ❑ Router (7 data flows @2.8GB/s)
- ❑ Torus Link
  - Light, low level “word stuffing” protocol
  - Diagnostic Messages for Fault-Awareness
  - Virtual Channel for deadlock-free transmission
  - 8b10b encoding and CRC

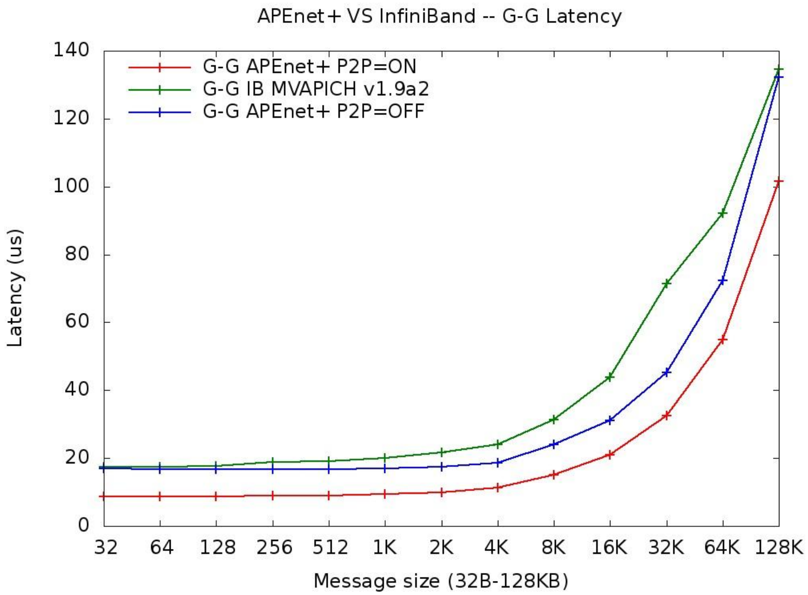




P2P on APEnet+

- ❑ P2P between Nvidia Fermi and APEnet+
  - First non-Nvidia device supporting it!!!
  - Joint development with Nvidia
  - APEnet+ board acts as a peer
- ❑ No bounce buffers on host. APEnet+ can target GPU memory with no CPU involvement
- ❑ GPUDirect allows direct data exchange on the PCIe bus
- ❑ Real zero copy, inter-node GPU-to-host, host-to-GPU and GPU-to-GPU
- ❑ Latency reduction for small messages





- ❑ Latency
  - APEnet+ G-G latency is lower up to 128KB
  - APEnet+ P2P latency ~8.2 μs (WR)
  - APEnet+ staging latency ~16.8 μs
  - MVAPICH/IB latency ~17.4 μs
- ❑ P2P=OFF
  - `cudaMemcpyD2H/H2D()` on host bounce buffers
  - Buffers pinned with `cuMemHostRegister`
  - `cuMemcpy()` ~10 μs

Use Cases

- **QUonG**:(Quantum chromodynamics ON Gpu) is a comprehensive initiative aiming at providing a hybrid, GPU-accelerated x86_64 cluster with a 3D toroidal mesh topology, able to scale up to $10^4 \div 10^5$ nodes
  - 16 nodes equipped with an APEnet+ board (4x4x1)
    - Intel Xeon E5620 double processor
    - 48 GB System Memory
    - 2 S2075 NVIDIA Fermi GPU
- APEnet+ customization: add specific I/O Interface and accelerators in FPGA
  - **NaNet**: Low Level Trigger GPU-based for HEP collider
    - Problem: reduce communication latency and its fluctuations.
      - Directly injecting data from the APEnet+ FPGA-based NIC into the GPU memory with no intermediate buffering, implementing GPUDirect RDMA.
      - Adding to the logic an offloading engine to manage a network stack protocol (UDP Offloading Engine) to avoid OS jitter effects.
  - **EURETILE**, Brain simulation: dedicated network for high speed connectome simulation (DPSNN model)
    - Partial re-design of the current DNP implementation
  - **APE3net:** Distributed read-out for the KM3 Neutrino Telescope
    - Deterministic latency
  - Low latency coupling of read-out system and GPU computing for E-ELT (European Large Telescope) and for X-Ray microscopes imaging (LBNL) (under evaluation)
    - 10 GbE interface coupled with GPU direct

Outline

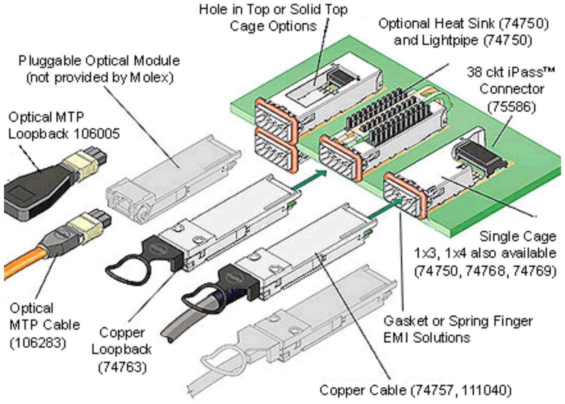
- ❑ Presentation of the APE project
- ❑ APEnet+ overview
- ❑ Description of the data transmission system
- ❑ The transmission control logic
- ❑ Conclusions

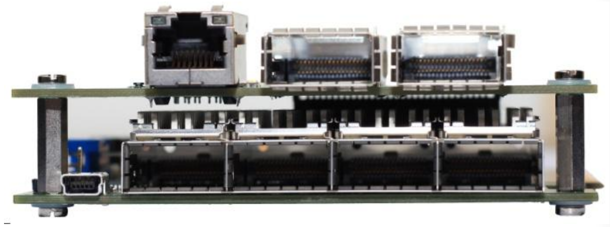
APElink Overview

- QSFP+ (Quad Small Form-factor Pluggable)
  - Hot-pluggable transceiver
  - 4 transmit and 4 receive lanes
  - Copper or optical medium
  - 40 Gbps, full bidirectional mode

- 6 channels on APEnet+ board
  - 4 modules on board, 2D torus topology, one I/O slot wide
  - 2 additional modules on a piggy-back, 3D torus topology, 2 slot wide card

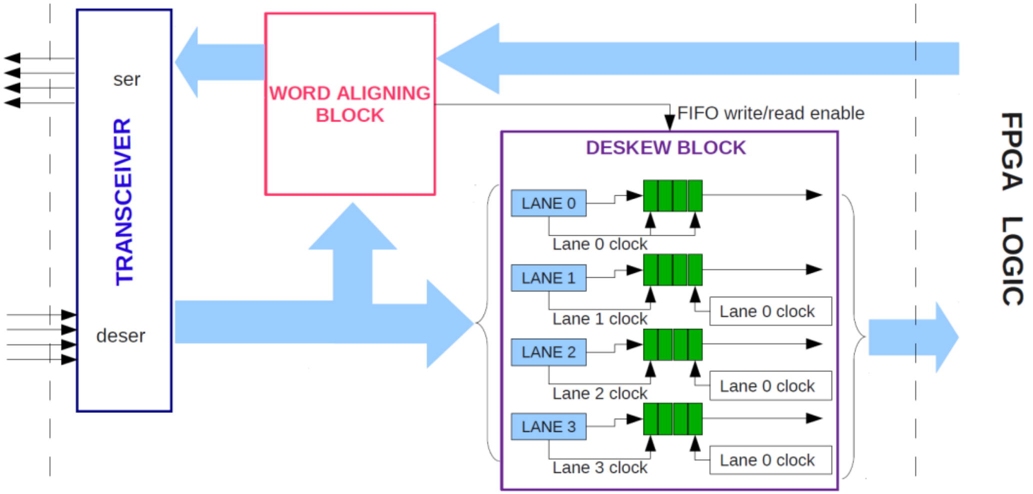
- Altera Stratix IV features:
  - 32 full duplex transceiver with CDR up to 8.5 Gbps
  - Physical Coding Sublayer (PCS) and Physical Medium Attachment (PMA)
  - each QSFP lane is connected to the transmit and receive side of the FPGA embedded transceivers (24 of 32)
  - aggregated bandwidth of 34 Gpbs per direction


Hole in Top or Solid Top Cage Options
Optional Heat Sink (74750) and Lightpipe (74750)
Pluggable Optical Module (not provided by Molex)
38 ckt iPass™ Connector (75586)
Optical MTP Loopback 106005
Single Cage 1x3, 1x4 also available (74750, 74768, 74769)
Optical MTP Cable (106283)
Copper Loopback (74763)
Gasket or Spring Finger EMI Solutions
Copper Cable (74757, 111040)



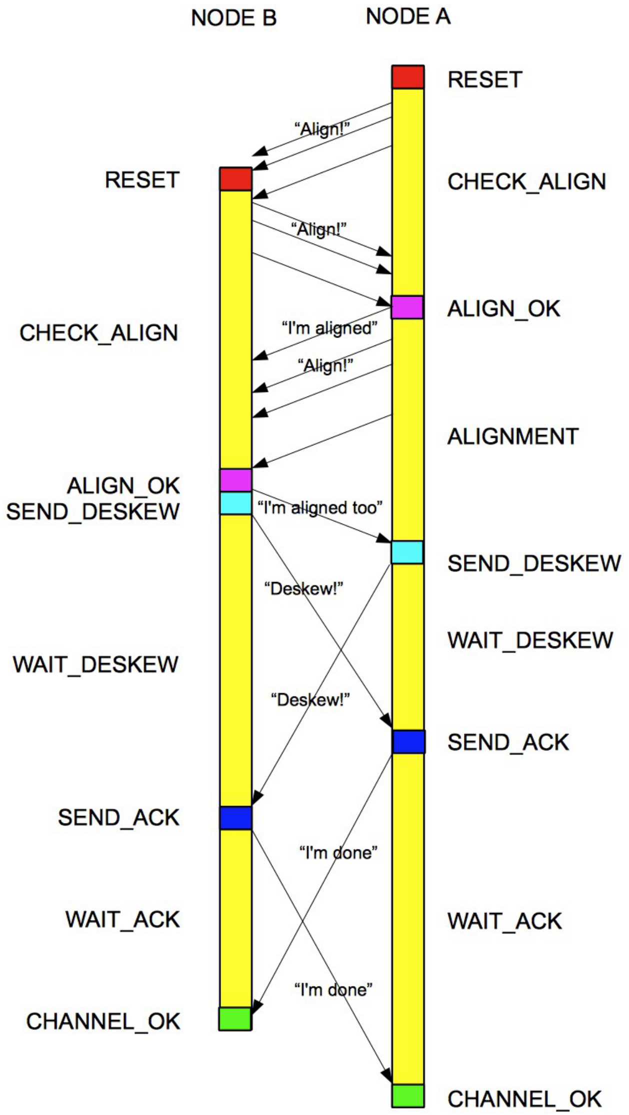
Aligning Custom Logic Block

- Transceiver Channel Bonding is guaranteed up to 4 lanes by Altera
- PCS level: 8B10B encoding for DC balancing in the serial data transmitted
  - special character for alignment purposes
- Word Alignment Block:
  - It receives parallel data from the deserializer
  - It restores word alignment based on special patterns received during link synchronization
- Deskew Block:
  - 4 DESKEW FIFOs, one per each channel lane
    - Write clock: specific clock provided by the transceiver.
    - Read clock: is the same for all FIFOs (clock is recovered by Lane 0)

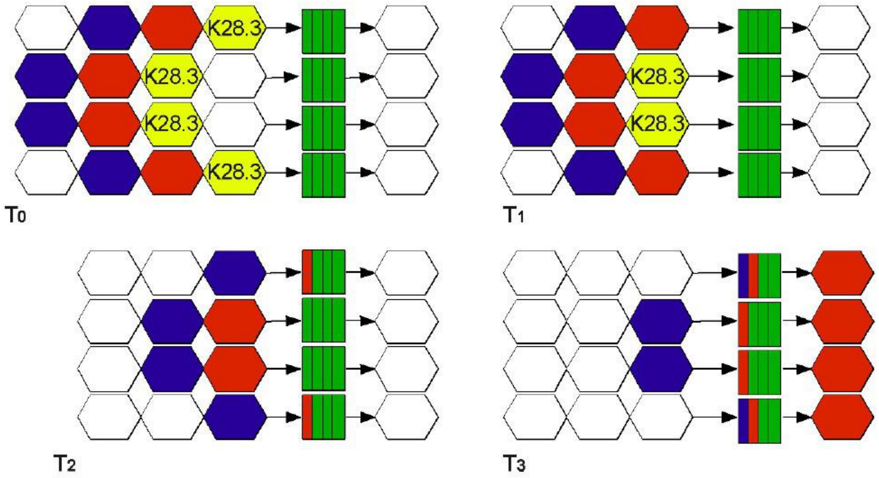


APElink Synchronization Procedure

- ❑ CHECK ALIGN phase:
  - TX: Each Word Aligning Block sends the alignment pattern “ALIGN WORD”
  - RX: It looks for the matching synchronization word in the received data stream.
- ❑ ALIGNMENT PROCEDURE result:
  - “ALIGN WORD” **is** received:
    - Synchronization is acquired by sending the “ALIGN OK WORD”,
  - “ALIGN WORD” **is not** received:
    - “CHECK ALIGN” phase restarts after the “RESTART ALIGN” state.
- ❑ channel alignment is assured when both nodes receive the “ALIGN OK WORD”
- ❑ DESKEWING PROCEDURE starts



Deskewing Procedure Example

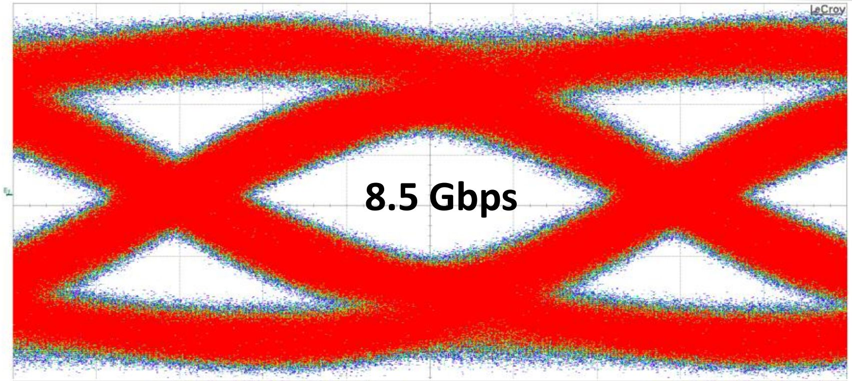


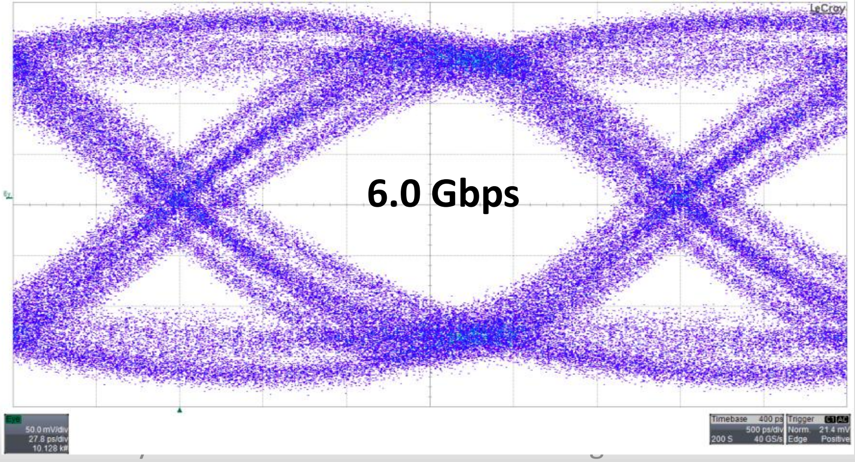
- ❑ The transmitter simultaneously sends a "DESKEW WORD (K28.3)" over all the four lanes.
- ❑ FIFO DESKEW
  - Write Enable is asserted after recognition of "DESKEW WORD" pattern
  - Read Enable is asserted when all FIFOs are no longer empty
- ❑ T0: the align command is received by two out of four lanes (say 0 and 3);
- ❑ T1: the special character is removed from lane 0 and 3, while align command is received by lane 1 and 2;
- ❑ T2: lane 0 and 3 write first word in FIFO, while special character is removed from lane 1 and 2;
- ❑ T3: lane 0 and 3 write second word, while lane 1 and 2 write first word in FIFO, and concurrently first word is read along all 4 lanes.

Bit Error Rate

BER	Direction	Cable Len	Data Rate
< 7.04 E-11	X+	5m	8.5 Gbps
< 8.12 E-11	Y+	5m	8.5 Gbps
< 7.04 E-10	Z+	5m	8.5 Gbps
< 9.80 E-13	X+	5m	7 Gbps
< 2.46 E-14	X+	2m	7 Gbps

- Fine tuning of PMAs analog settings
- Manual tuning: *trial and error* procedure
- Altera transceiver toolkit can dynamically reconfigure PMAs settings:
  - TX: Pre-emphasis and DC GAIN
  - RX: Equalization and VOD
- TESTBED FIRMWARE
  - TX: random signal generator (PRBS @32bits)
  - RX: data checker



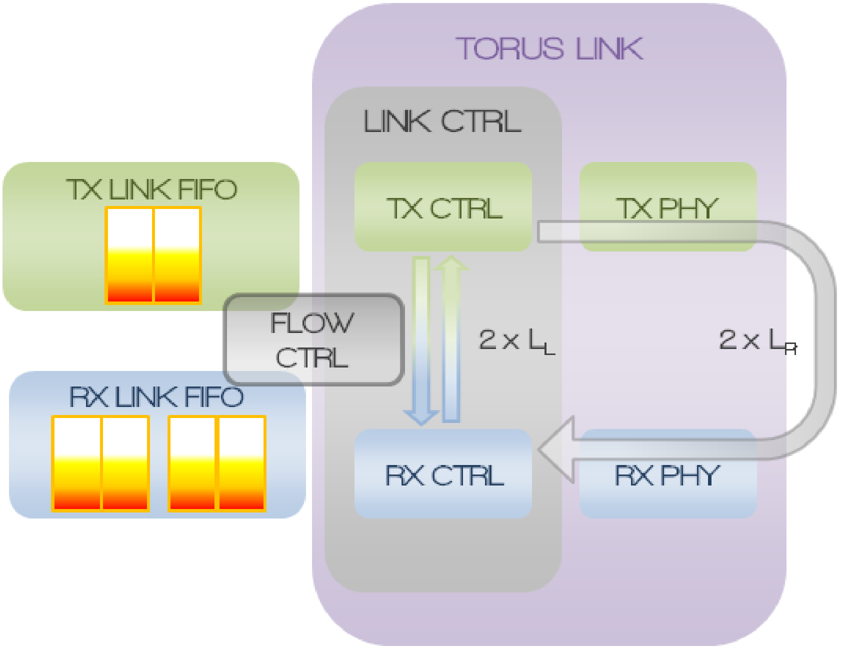


Outline

- ❑ Presentation of the APE project
- ❑ APEnet+ overview
- ❑ Description of the data transmission system
- ❑ The transmission control logic
- ❑ Conclusions

Data-Link Layer

- ❑ Managing the data flow
- ❑ Encapsulating packets into a light, low-level word stuffing protocol
- ❑ Detecting trasmission error via CRC
- ❑ Implementing virtual channels to guarantee deadlock free routing
- ❑ Diagnostic Messages for Fault-Awareness



Performance Analysis

- ❑ 8B10B encoding: 20% loss of bandwidth
- ❑ APEnet Data Packets:
  - Header (128-bit word)
  - Payload (Max Size, S_{MAX} = 4KB, S_{MAX} = 256 cycles)
  - Footer (128-bit word)
- ❑ A communication cannot be initiated if the receiver does not have sufficient space to accomodate the entire message (wormhole switching)

- ❑ Three Efficiency Factors:
  - E_1: related to the adopted protocol
  - E_2 : status information (receiving FIFO occupancy and diagnostic messages)
  - E_3 : data flow management

E_1: Protocol Efficiency Factor

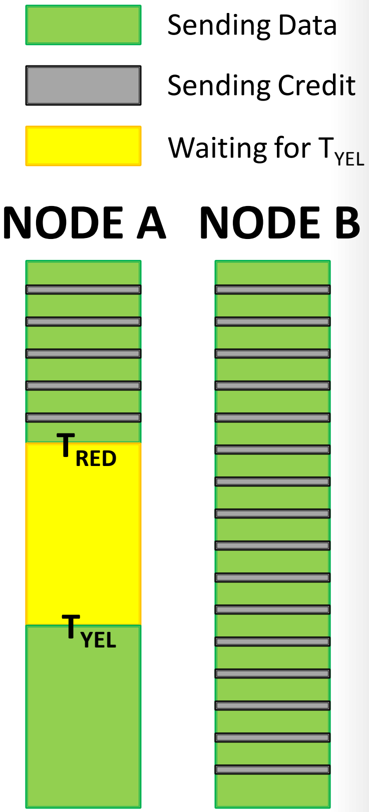
- Word Stuffing Protocol: *Magic* (128-bit word) is used to distinguish *control* and *data* frame
- *Start* (128-bit word): initialization of a new data packet

- P = 64 bytes (*Magic, Start, Header* and *Footer*)

$$\boxed{E_1 = \frac{S_{MAX}}{P + S_{MAX}} = \frac{4096}{64 + 4096} = 0.985}$$

Typical Data Flow and E_2

- *Credit*: 128-bit word containing the status of the receiving FIFOs



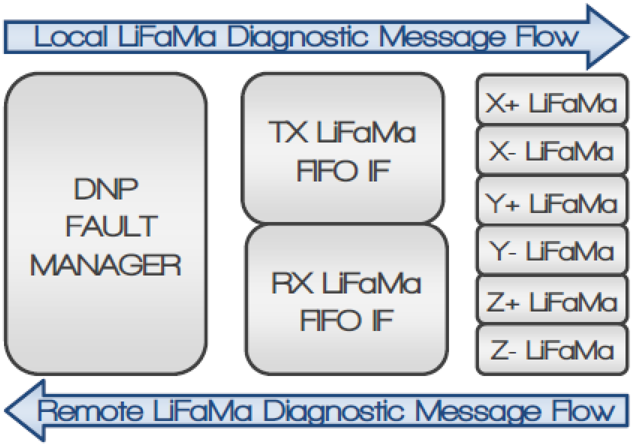
- Node A and Node B transmit data simultaneously
- The transmitter keeps track of the sent data
- Node A reaches T_{RED}. It stops sending data
- Node A ensures that the receiving FIFOs have been emptied at least up to value T_{YEL}
- Node A restarts sending data

- Node B could be committed sending data over 256 cycles!!
- To avoid latency addition the transceiver sends a Credit every C cycles (C=35)

$$\boxed{E_2 = \frac{C}{C+2} = 0.946}$$

APElink Diagnostic Messages

- ❑ LO|FA|MO (Local Fault Monitor)
  - An approach to Fault Awareness for Distributed Systems
  - Mutual watchdog mechanism between host and DNP on each node
  - No single point of failure!


Local LiFaMa Diagnostic Message Flow

DNP FAULT MANAGER

TX LiFaMa FIFO IF

RX LiFaMa FIFO IF

X+ LiFaMa

X- LiFaMa

Y+ LiFaMa

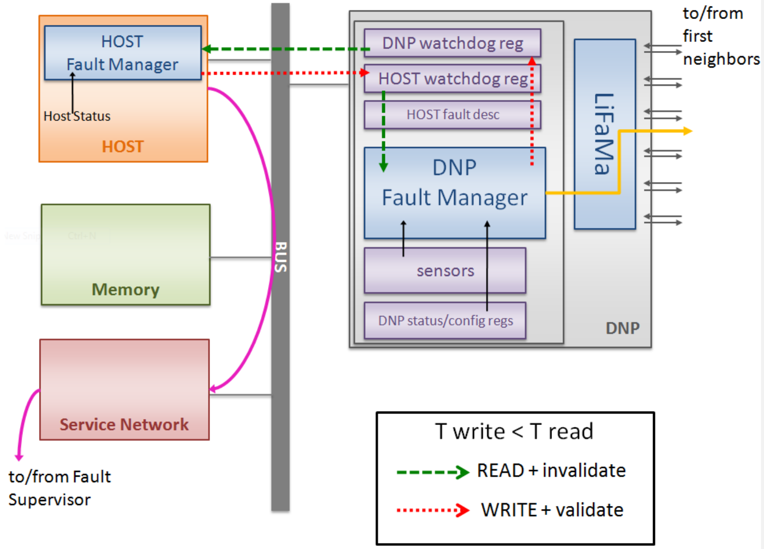
Y- LiFaMa

Z+ LiFaMa

Z- LiFaMa

Remote LiFaMa Diagnostic Message Flow

- ❑ Diagnostic Messages are embedded into the credit without affecting the performance
- ❑ A credit every 35 cycle @28gbps means an update frequency of ~200ns


HOST Fault Manager

Host Status

HOST

Memory

Service Network

to/from Fault Supervisor

BUS

DNP watchdog reg

HOST watchdog reg

HOST fault desc

DNP Fault Manager

sensors

DNP status/config regs

DNP

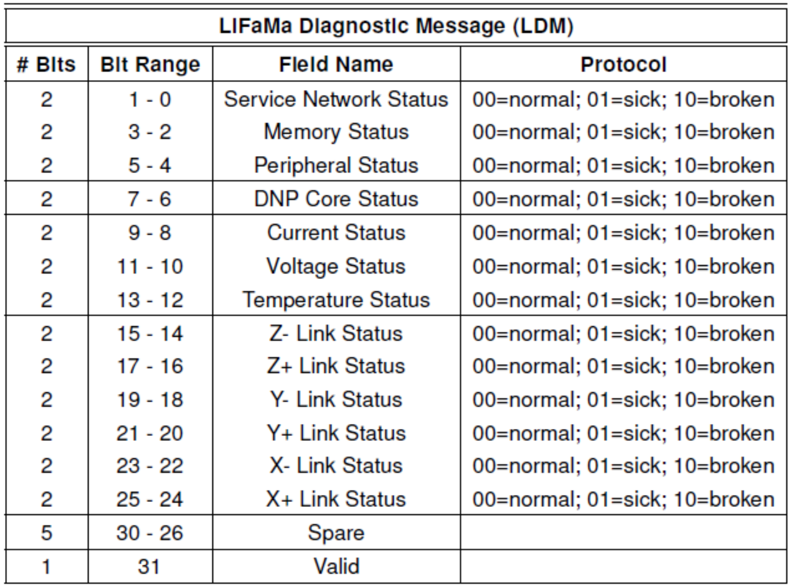
LiFaMa

to/from first neighbors

T write < T read

READ + invalidate

WRITE + validate



LIFaMa Diagnostic Message (LDM)			
# Bits	Bit Range	Field Name	Protocol
2	1 - 0	Service Network Status	00=normal; 01=sick; 10=broken
2	3 - 2	Memory Status	00=normal; 01=sick; 10=broken
2	5 - 4	Peripheral Status	00=normal; 01=sick; 10=broken
2	7 - 6	DNP Core Status	00=normal; 01=sick; 10=broken
2	9 - 8	Current Status	00=normal; 01=sick; 10=broken
2	11 - 10	Voltage Status	00=normal; 01=sick; 10=broken
2	13 - 12	Temperature Status	00=normal; 01=sick; 10=broken
2	15 - 14	Z- Link Status	00=normal; 01=sick; 10=broken
2	17 - 16	Z+ Link Status	00=normal; 01=sick; 10=broken
2	19 - 18	Y- Link Status	00=normal; 01=sick; 10=broken
2	21 - 20	Y+ Link Status	00=normal; 01=sick; 10=broken
2	23 - 22	X- Link Status	00=normal; 01=sick; 10=broken
2	25 - 24	X+ Link Status	00=normal; 01=sick; 10=broken
5	30 - 26	Spare	
1	31	Valid	

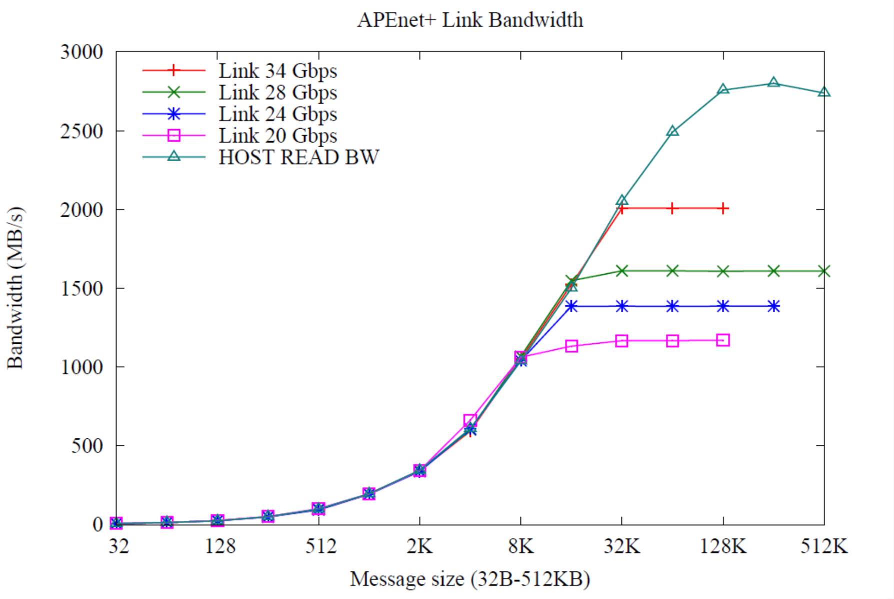
E_3: Data-Flow Efficiency Factor

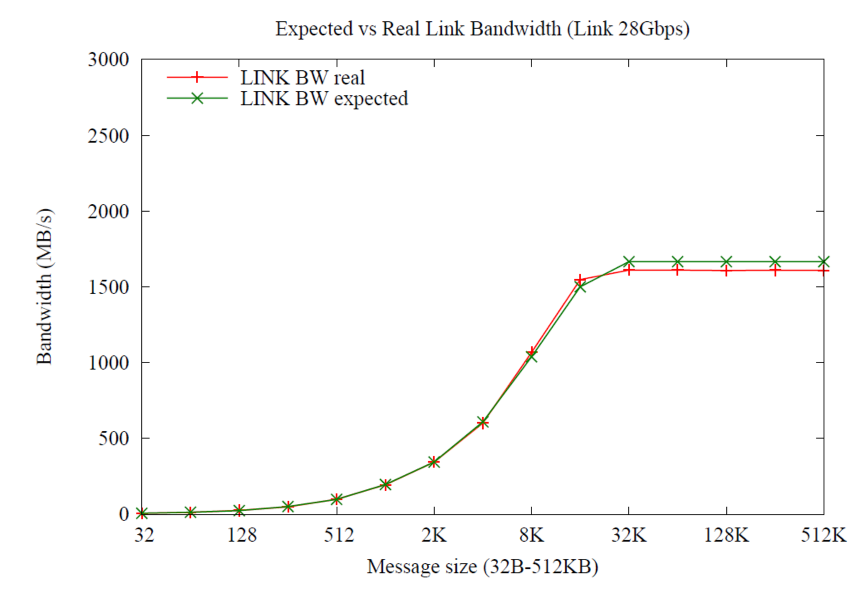
- Evaluation of the data interruption cycle
  - Flight time of Last Sent Data and Credit (Remote Latency = 35 cycles)
  - Exchanging information between receiving and trasmitting side of the same node (Local Latency = 20 cycles)
    - RX clock is reconstructed from the incoming data (CDR)
    - TX clock is locally generated by PLLs
    - Synchronization is essential
  - In the worst case the receiving node takes C cycles to submit the proper Credit
- Waiting for W=145 Cycles to assure that the Credits are updated
- **REMEMBER: WORMHOLE ROUTING!!!**

$$\boxed{E_3 = \frac{\lfloor \frac{T_{RED}}{S_{MAX}} \times S_{MAX}}{(\lfloor \frac{T_{RED}}{S_{MAX}} \times S_{MAX}) + W}}$$

THEORY VALIDATION

- RX FIFOs 512x128bit
- Point-to-point bandwidth test



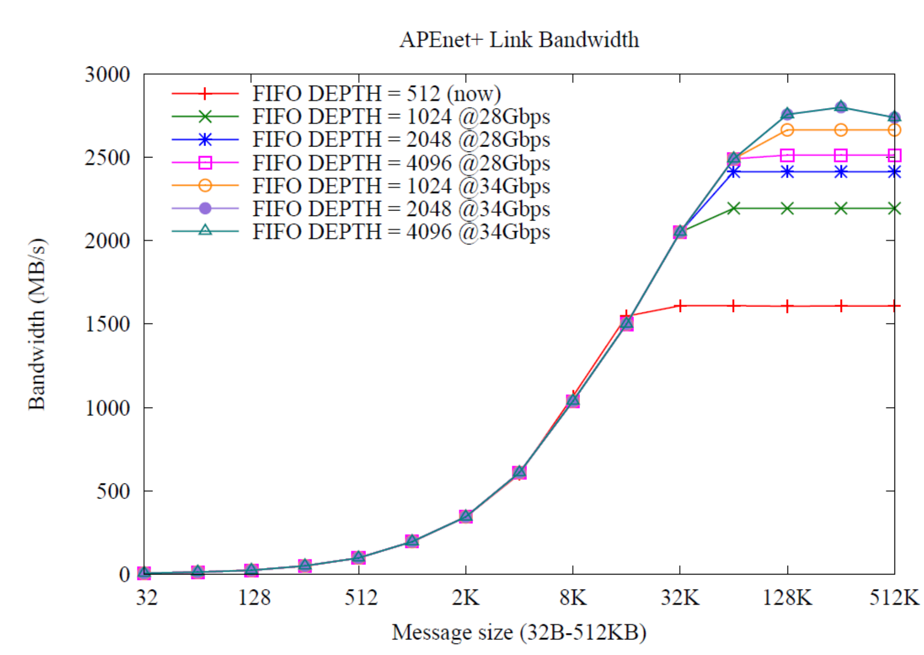


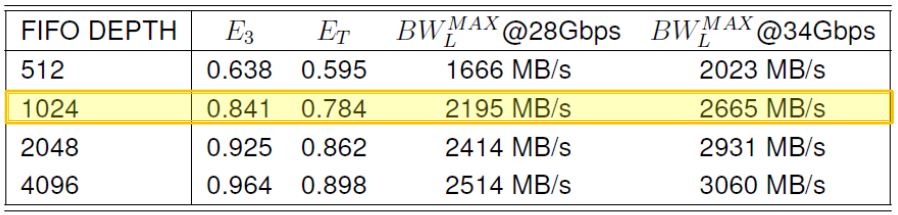
- If host read bandwidth BW_{READ} is less than the link maximum bandwidth

$$BW_{link} = BW_{host}$$

$$E_T = E_1 \times E_2 \times E_3$$

Performance vs Memory Resources





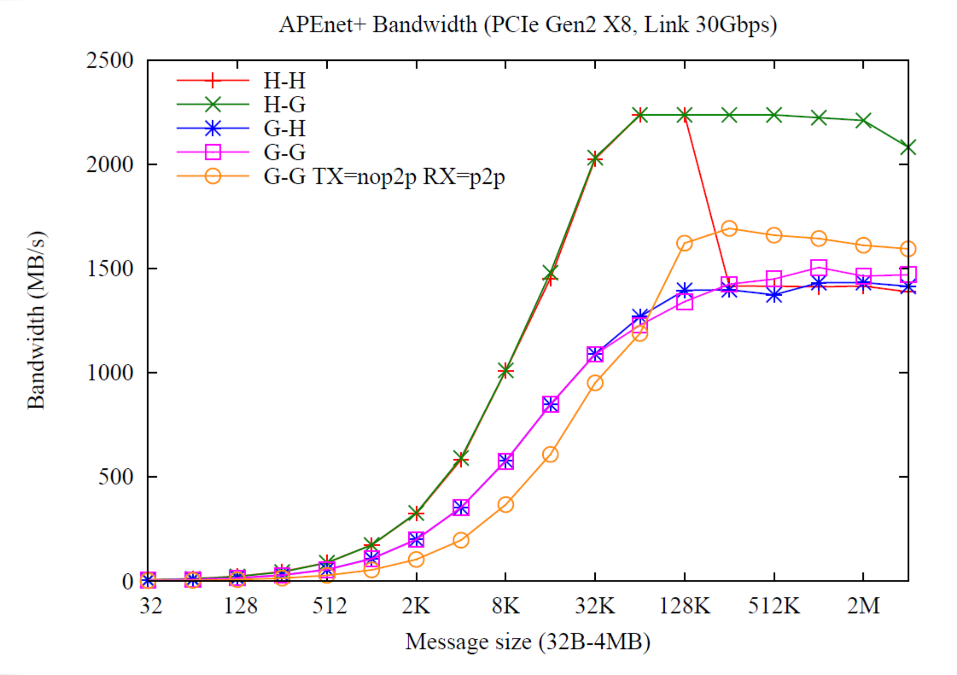
FIFO DEPTH	E_3	E_T	BW_L^{MAX}@28Gbps	BW_L^{MAX}@34Gbps
512	0.638	0.595	1666 MB/s	2023 MB/s
1024	0.841	0.784	2195 MB/s	2665 MB/s
2048	0.925	0.862	2414 MB/s	2931 MB/s
4096	0.964	0.898	2514 MB/s	3060 MB/s

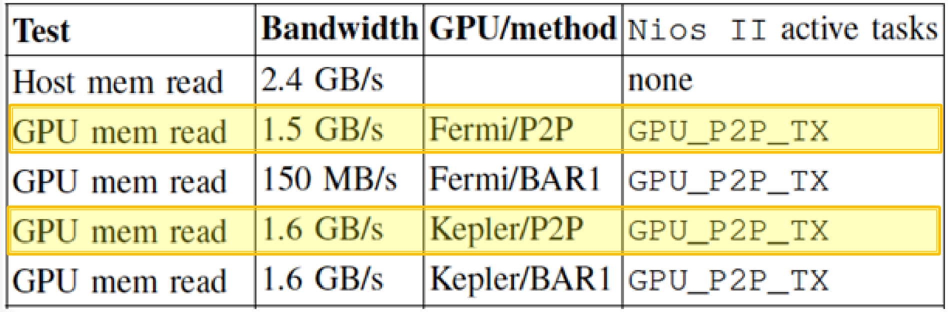
- ❑ 16KB RX FIFO (x2 considering Virtual Channels)
- ❑ 8KB TX FIFO
- ❑ 40 KB per channel
- ❑ Total Memory 240KB (Stratix IV provides 1.74MB, ~14% only)

Outline

- ❑ Presentation of the APE project
- ❑ APEnet+ overview
- ❑ Description of the data transmission system
- ❑ The transmission control logic
- ❑ Conclusions

Bandwidth Overview





Test	Bandwidth	GPU/method	Nios II active tasks
Host mem read	2.4 GB/s		none
GPU mem read	1.5 GB/s	Fermi/P2P	GPU_P2P_TX
GPU mem read	150 MB/s	Fermi/BAR1	GPU_P2P_TX
GPU mem read	1.6 GB/s	Kepler/P2P	GPU_P2P_TX
GPU mem read	1.6 GB/s	Kepler/BAR1	GPU_P2P_TX

- CPU Memory Read Bandwidth ~ 2.4 GB/s
  - Completely handled by Kernel Driver
  - 2 DMA channels implemented to reduce the latency between two consecutive PCI data read request
- GPU Memory Read Bandwidth ~ 1.5 GB/s
  - Unlimited prefetch window and read request accelerated Mechanism (a read req every 80ns)
  - Upper limit of NVIDIA P2P DMA Engine
  - Traditional Flow with Host bounce buffer is necessary for big size messages.
- APElink Bandwidth ~ 2.2 GB/s @350 MHz
  - Link Performance depends on the receiving fifo depth (16KB now)
  - TODO: Improving the signal integrity to increase the frequency up to 425MHz
- GPU Memory Write Bandwidth ~ 2.2 GB/s
  - TLB hardware
- CPU Memory Write Bandwidth
  - ~ 1.5 GB/s Handled by the Nios II microcontroller @200MHz
  - ~ 2.2 GB/s virtual to physical address translation managed in hardware (very memory demanding)

Future Works

- ❑ Improving the APElink Performance
  - Fine tuning of PMAs analog settings to reduce the BER @8.5Gbps
  - Increase the size of the receiving FIFOs (fast-and-dirty solution)

- ❑ DNP on next generation 28nm FPGA
  - PCIe GEN3 interface for x2 Host interface throughput (8GB/s vs 5GB/s)
  - Complete porting, test and performance analysis on ALTERA Stratix V Development Kit
    - Enhanced embedded transceiver (14.1 Gbps VS 8.5 Gbps)
    - More efficient data encoding scheme (128b/130b vs 8b/10b)
  - Design of the 3D link data physical channel using new FPGA embedded transceiver (56 Gbps)

- ❑ More interesting, analyzing Efficiency Factors
  - E_1, E_2 are very protocol dependent and their contribution is 7%
  - E_3 is mainly due to the interruption of data transmission waiting for an updated credit
  - Redefinition of Flow Control Logic: W term must be suppressed (How?)

$$\boxed{E_3 = \frac{\lfloor \frac{T_{RED}}{S_{MAX}} \times S_{MAX}}{(\lfloor \frac{T_{RED}}{S_{MAX}} \times S_{MAX}) + \cancel{W}}}$$

The End

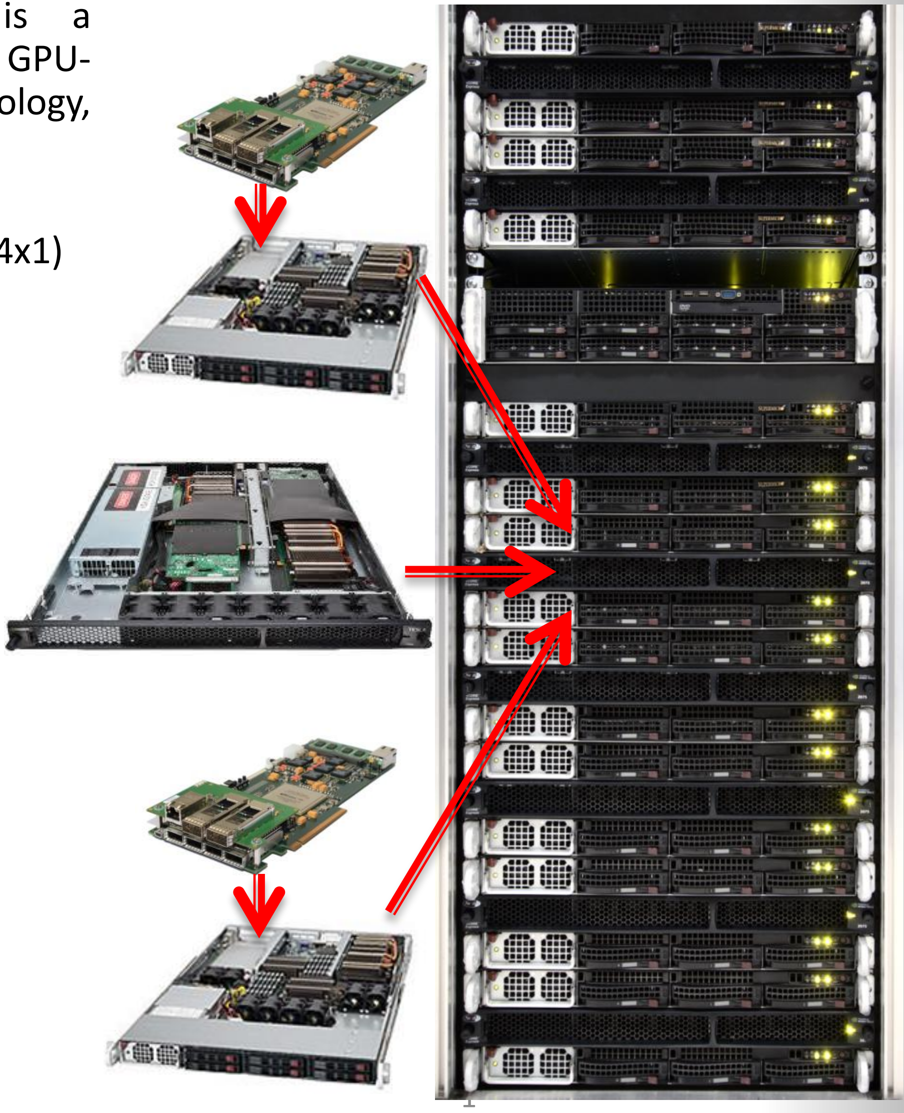
THANK YOU

BACK-UP SLIDES

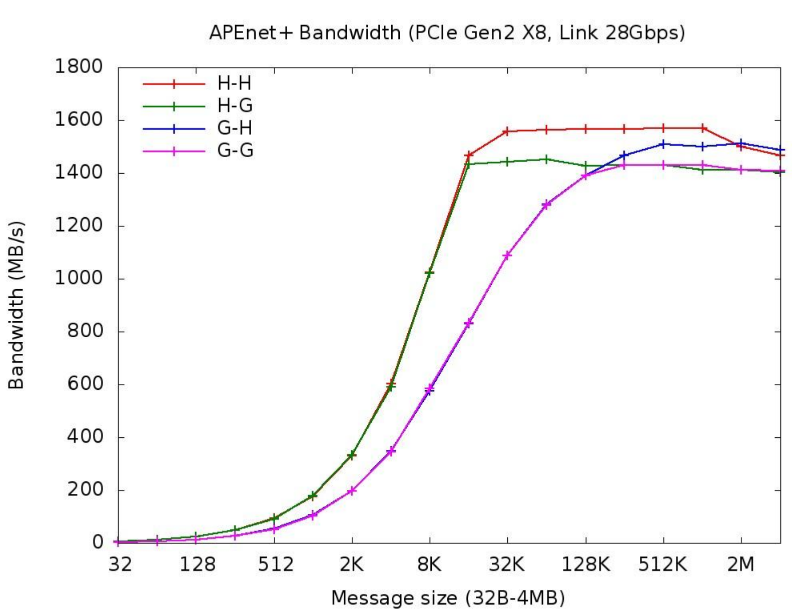
QUonG: EURETILE HPC platform

Quantum chromodynamics ON Gpu (QUonG) is a comprehensive initiative aiming at providing a hybrid, GPU-accelerated x86_64 cluster with a 3D toroidal mesh topology, able to scale up to $10^4 \div 10^5$ nodes

- Current Status:
  - 16 nodes equipped with an APEnet+ board (4x4x1)
- QUonG Hybrid Computing Node:
  - double Intel Xeon E5620 processor
  - 48 GB System Memory
  - 2 S2075 NVIDIA Fermi GPUs
  - 1 APEnet+ board
  - 40 Gb/s InfiniBand Host Controller Adapter
- QUonG Elementary Mechanical Unit:
  - 3U Sandwich (2 Vertexes on the APEnet+ 3d network):
    - 2 Intel dual Xeon servers
    - 4 NVIDIA Tesla M2075 GPU
- Software Environment:
  - CentOS 6.3
  - NVIDIA CUDA 4.2 driver and dev kit
  - OpenMPI and MVAPICH2 MPI available

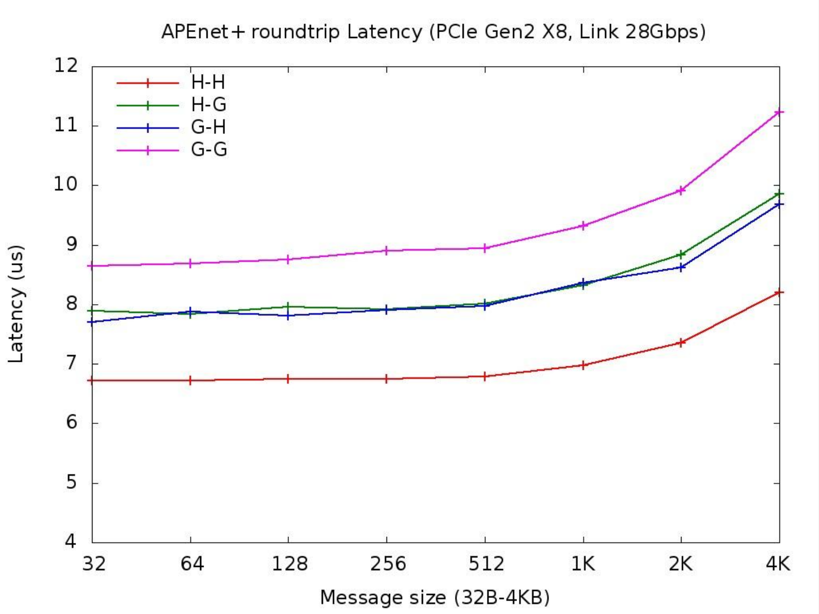


Latency & Bandwidth Overview



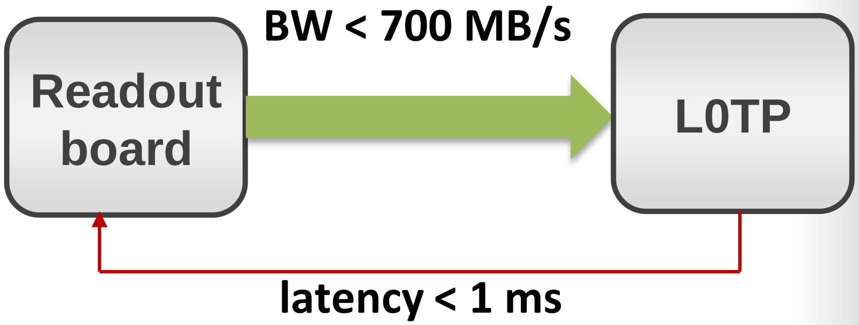
- ❑ Bandwidth:
  - Host RX ~1.6 GB/s
  - GPU RX ~1.4 GB/s (switching GPU P2P window before writing)
  - Limited by the RX processing (Accelerate the buffer research and virtual to physical translation performed by the Nios)
- ❑ GPU TX CURVES:
  - P2P read protocol overhead

- ❑ The latency is estimated as half the round trip time in ping-pong test
- ❑ ~ 8÷10 µs G-G latency
- ❑ World Record for G-G latency

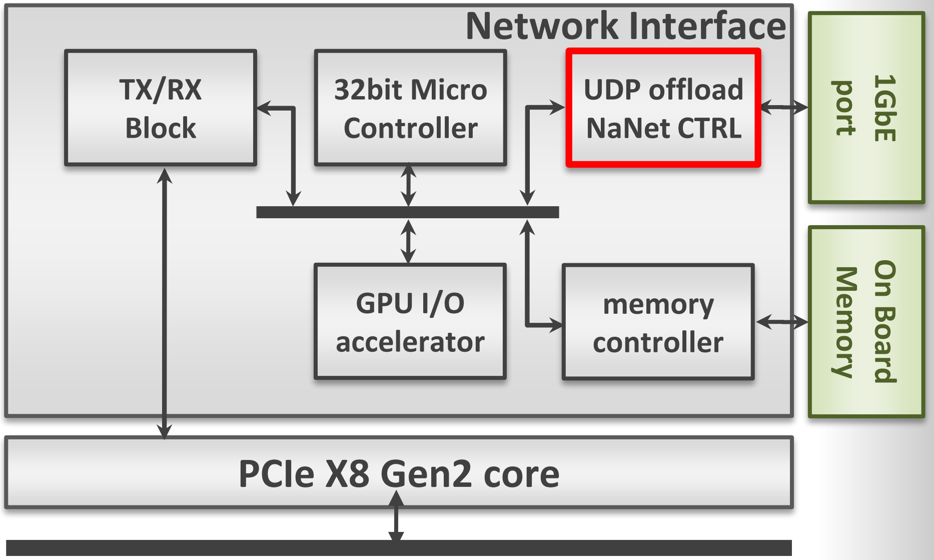


NaNet: APEnet + NA62 CERN Experiment

- ❑ GPU L0 TRIGGER for HEP Experiments
  - Implement a RO Board-L0 GPU link with:
  - Sustained Bandwidth < 700 MB/s, (RO board output on GbE links)
  - Small and stable latency



- ❑ Problem: reduce communication latency and its fluctuations. How?
  - Offloading the CPU from network stack protocol management
  - Injecting directly data from the NIC into the GPU(s) memory



- ❑ NaNet solution:
  - APEnet+ FPGA-based NIC with an additional network stack protocol management offloading engine (UDP Offloading Engine).