PXD DAQ Overview

Jens Sören Lange
Justus-Liebig-Universität Giessen
Belle II PXD DEPFET Workshop
Charles University, Prague, Czech Republic
January 26-27, 2010

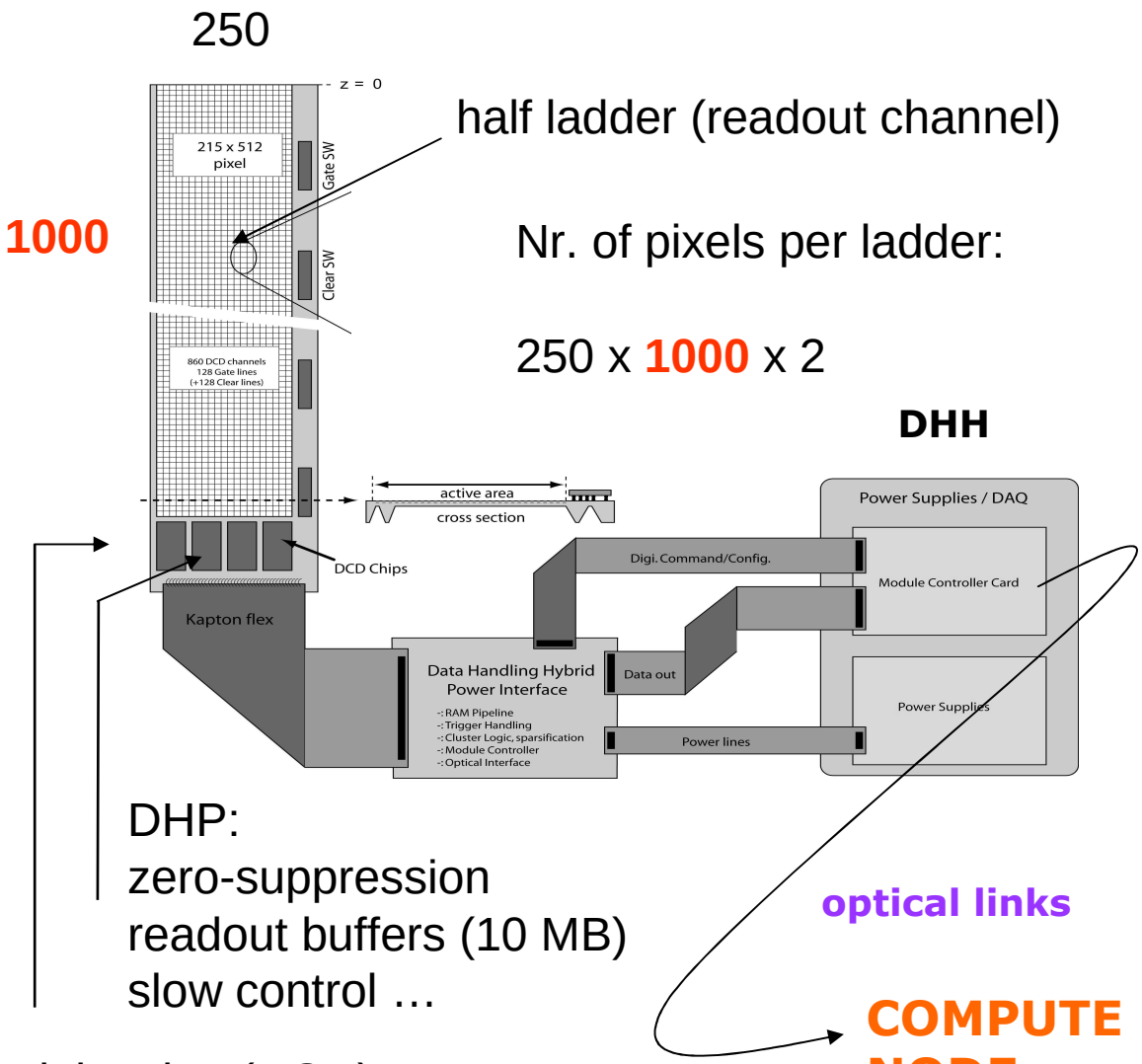
Numbers and Protocols for the DAQ part of the TDR

Content

- Trigger
  - Hardware trigger
  - Filter Algorithms for FPGA
  - Protocol (for TDR)
- DAQ
  - Event building
  - SVD parasitic readout for PXD
- Status of Compute Node
  - Performance tests of new board
  - News from IPMI

Glossary

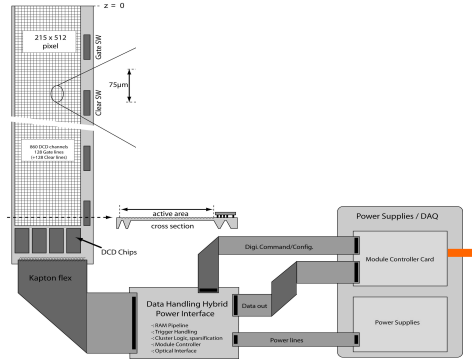
- CN = compute node
- GB, MB, kB = gigabytes, megabytes, kilobytes
- Gb, Mb, kb = gigabits, megabits, kilobits



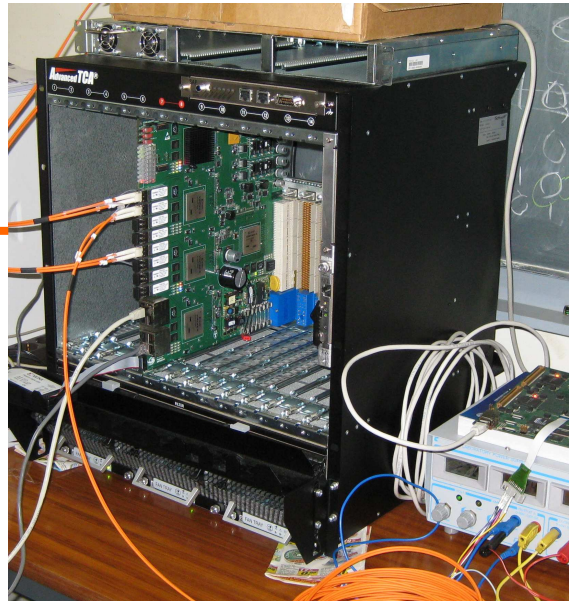
- 40 half ladders
- 0.5 Mill. Pixel per half ladder
- 10 Million pixels (px) total
- assumed accupancy 1%
  (cluster size = 2 px,
  readout time 20 us)
  10^5 pixels fired in each frame
- trigger:
  every 10th frame is read out
  trigger rate 30 kHz
- 3 x 10^9 px/s
- 4 bytes per px (pos + ADC)
- 12 GB/s total
  40 optical links
  300 MB/s per 1 optical link
  2.4 Gb/s per 1 optical link
- we must assume
  peak occupancy
  2% (i.e. factor 2 safety margin)

PXD Backend Readout: ATCA System

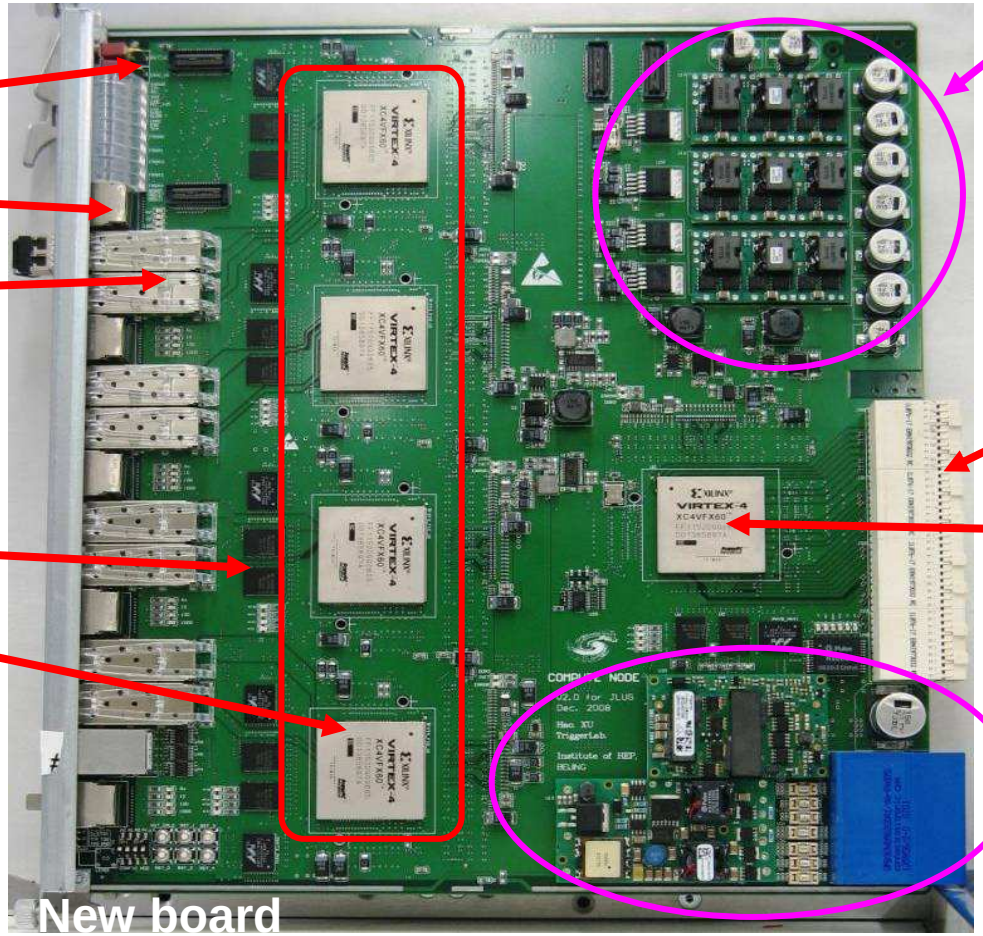
„IHEP/Gießen Box"
ATCA based Compute Nodes
Virtex-4 FPGAs



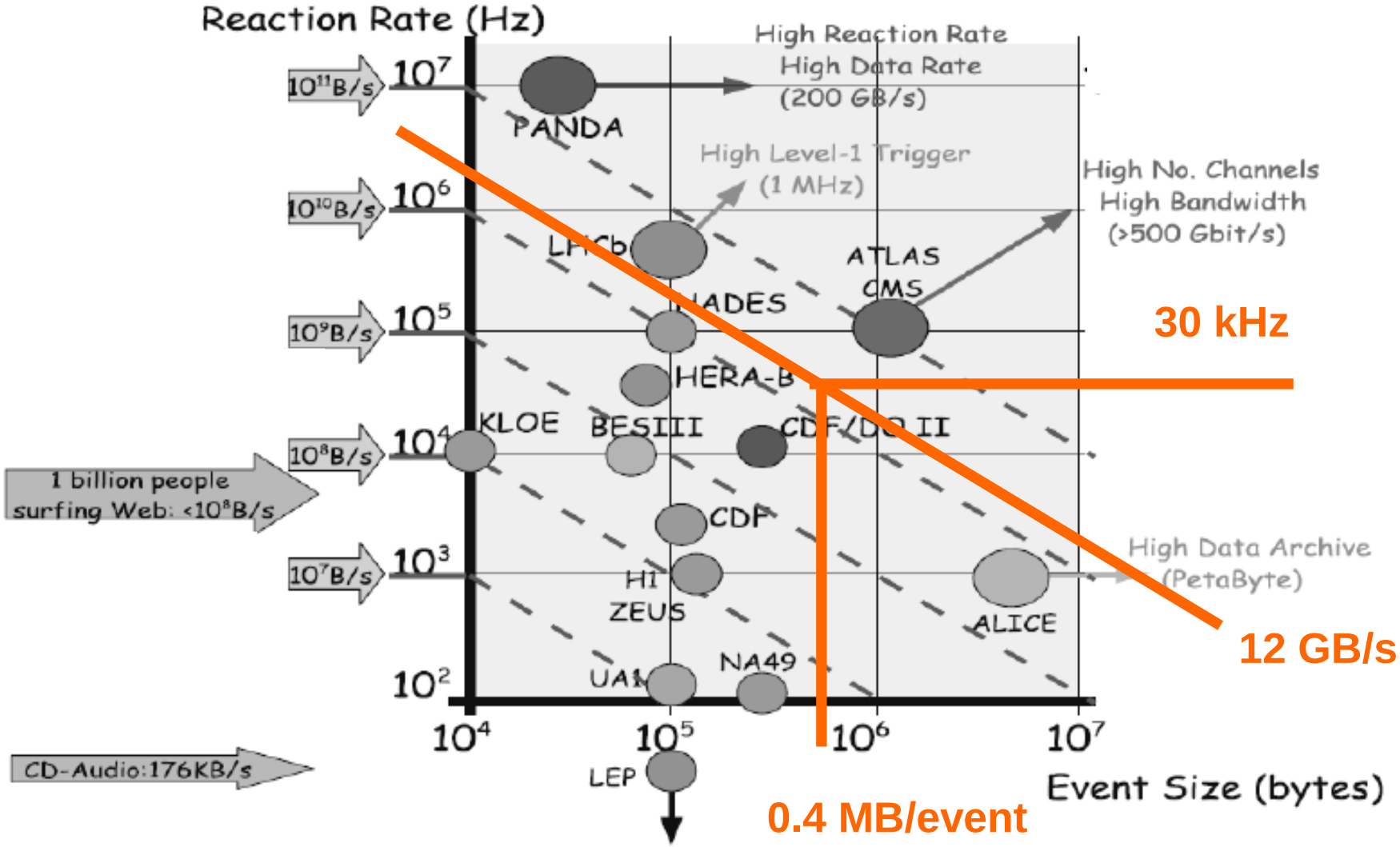
Optical links
// 40



Compute Node PCB



Bandwidth and Event Size Comparison



Trigger

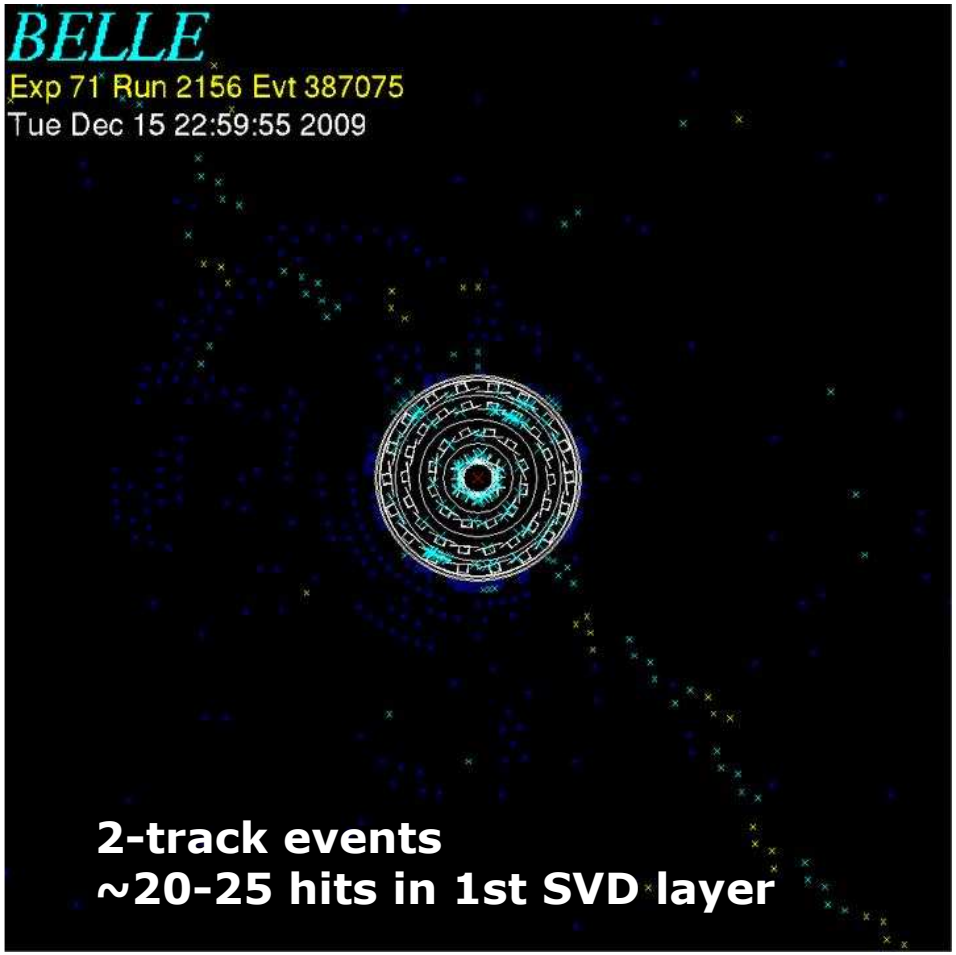
PXD Trigger

- There are 3 triggers
  1. Hardware trigger
     by Belle II Global Decision Logic
     reduction factor $f_1 \simeq 10$
     (i.e. only every ~10th frame is sent to readout)
     must arrive within 2-3 us
     requires a complete PXD trigger system and timing synchronisation
  2. Track finder algorithm on FPGA in ATCA system
     using only PXD and SVD (2+4 hits)
     reduction factor f_2
  3. Track finder algorithm on L3 farm („HLT")
     sent to ATCA system, distributed to each FPGA
     track extrapolation to PXD hits > reduction factor f_3
     > FPGAs must buffer and wait for ~5 seconds
     HLT might be GPU based, see slides by Katayama-san
- Product reduction factor f_2 x f_3 must be order of 10 (or more)
  why?

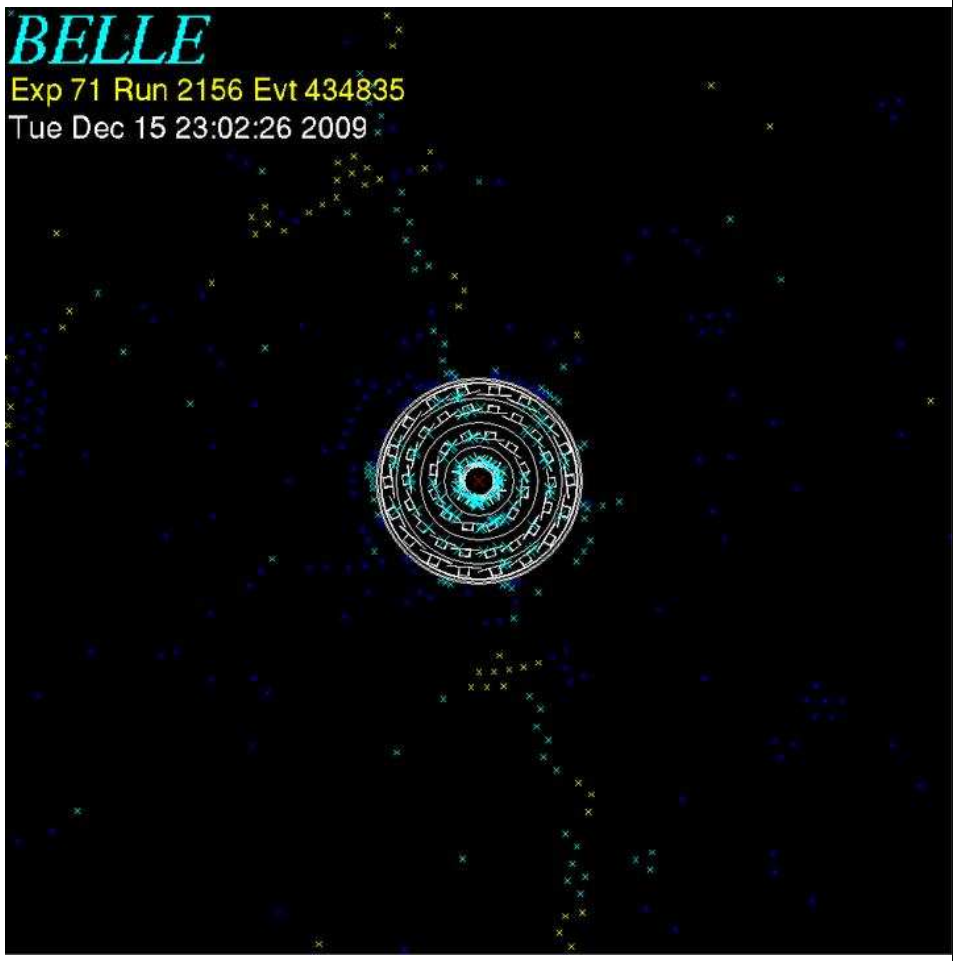
Which reduction factors must be achieved?

- Possible bottleneck is output of ATCA system to EVB farm by GB Ethernet
  - at 1% occupancy, f_1=10, f_2 x f_3=10
    raw data is 12 GB/s
    output data to event builder is 1.2 GB/s
  - distributed on 40 Gb ethernet links
    $\simeq$ 30 MB/s per 1 Gb ethernet link
  - acceptable, but
    - CPU maybe busy by interrupt handling
    - PXD data might saturate EVB farm
    - safety margin up to occupancy 2% (60 MB/s) needed
  - achieved $\simeq$ 25 MB/s on PowerPC (on FPGA)
    theoretical limit is 125 MB/s
- **we have to achieve reduction factors higher than 10**

What is the goal of the Filter Algorithm?



2-track events
~20-25 hits in 1st SVD layer



What does the FPGA filter algorithm need to do?

2 primary tasks:

1. delete events
   (e.g. QED background)

2. clean up events
   i.e. discard part of the event
   (e.g. discard PXD hits inside physics events
   e.g. from synchrotron photons)

Algorithm #1: C. Heller, C. Kiesling, A. Moll (MPI)
Algorithm #2: D. Münchow, S. L. (Gießen)

Filter Algorithms: Track Finder

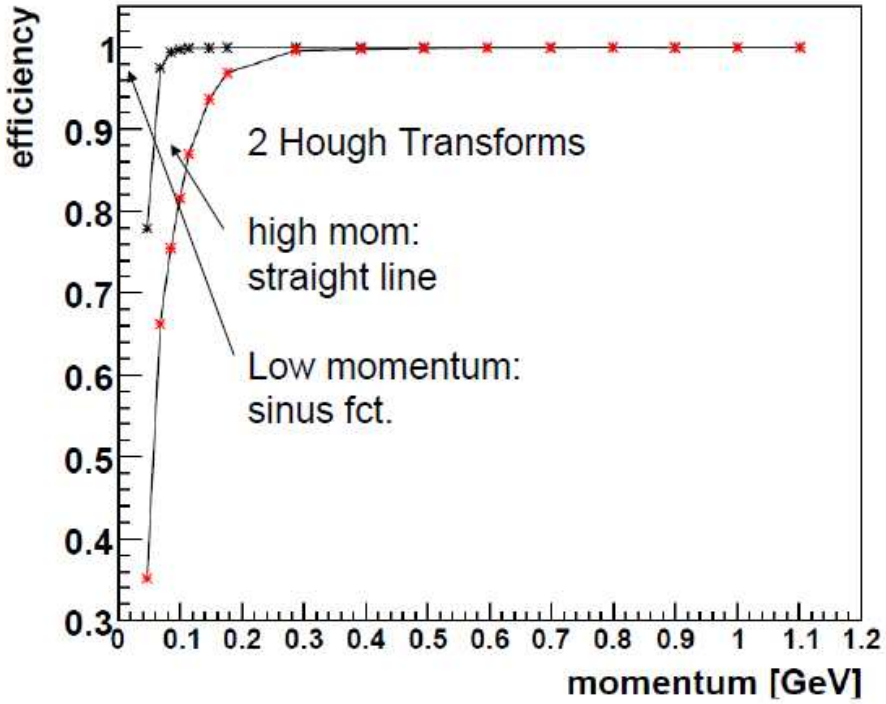
- to be programmed in VHDL for FPGA
- so far 2 approaches
  - Algorithm #1:
    2-dim track search in ϕ sectors
    Hough transform in rϕ real space
    2 different transforms for low and high momentum
    Algorithm #2:
    3-dim helix tracking with look-up tables
    (adjusted to 1 MB only free blockRAM on FPGA)
    conformal map (circle $\rightarrow$ straight line)
    2 x Hough transform
    (rϕ in conformal space, z in real space)

Algorithm #1: C. Heller, C. Kiesling, A. Moll (MPI)
Algorithm #2: D. Münchow, S. L. (Gießen)

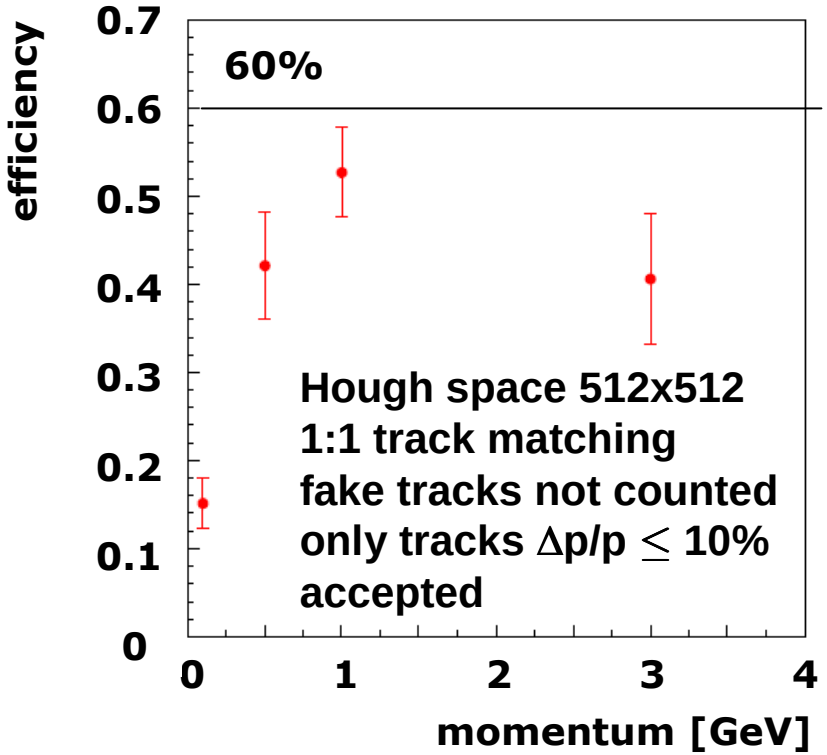
Efficiency

http://www-linux.gsi.de/~lange/sbelle/thesis_david_muenchow.pdf

Algorithm #1



Algorithm #2



see slides by Andreas Moll

Trigger Protocol for TDR, Filter Algorithm YES/NO Decision

- PXD system will **not** deliver a trigger (back) to the DAQ system
  PXD system will just delete data.
- for deleted PXD events, it might happen,
  that all the other detectors have a valid event
  > what to do with offline events w/o PXD data?
  > write a downscaled fraction of these to tape
- if hardware trigger issues NO
  > data will never reach the optical links
  (overwritten in DHP buffer)
- If FPGA algorithm issues a NO, there are 2 possible protocol schemes
  a.) send nothing > EVB waits and runs into a TIMEOUT
  b.) send empty frames > generates overhead
  assume in worst case 1 empty GB frame
  9.6 kB x 30 kHz x 90% (factor 10) = 2.6 MB/s
  > we prefer solution B
- for cleaned events, we have to mark,
  which part of the events is cleaned
  proposal: a bitmask
  e.g. 40 bytes = 125 pixels in z direction

Hardware Trigger

- We need f_1=10 by a GDL signal within 2-3 us, otherwise optical links are overloaded

- Consequences:
  - trigger system required (distribute GDL output)
  - timing synchronisation required of RF/5 (DHP clock) with RF/16 (beam REVO signal, reference for Belle II) RF = 508,89 MHz
  - latch required in the 70's: all trigger cables have same length today: latching trigger signal until SYNC signal arrives

  these are probably 3 more hardware projects (PCBs)

  - issue for MC simulations: several GDL inputs have p_T cut (e.g. 300 MeV).

Trigger Protocol for TDR, Hardware Trigger

- Rest of Belle II DAQ is pipelined
- PXD pipelining?
  - what happens, if another trigger arrives, while DHP readout is in progress
    > does PXD generate BUSY signal?
  - what happens, if a trigger arrives for an event, which is not the first in the DHP buffer?
    DHP has 10 MB buffer, $\simeq$ 25 events
    > we need to flush the pipe
    > generates data overhead
    (untriggered events in stream)

DAQ

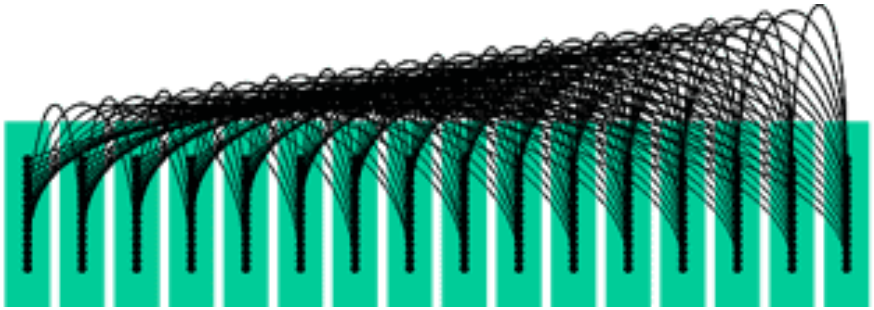
What is PXD DAQ?

- 1. Receive PXD data
- 2. Receive SVD data
- 3. **Event building**
- 4. Filter algorithm on FPGA
  (track finding with PXD+SVD)
- 5. Sending PXD subevent data to EVB farm
  (if filter algorithm issues no,
  then data are not sent)

Event Building

- For PXD only:
  - 30 kHz, i.e. 30,000 trigger numbers in 1 sec
  - events are fragmented in 40 pieces on 40 FPGAs
  - assumption: buffer for 5 seconds
    (2 GB memory, 3 kHz event rate after hardware trigger,
    0.4 MB event size)
  - There are 3,000 x 5 x 40 data fragments
    in the memory at the same time
    which need to be sorted (heapsort, quicksort, …)
    and combined to real events
    = 0.6 Million fragments
- large combinatorics (0.6 Mill. X 0.6 Mill.)$^{40-1}$
- requires synchronisation between data fragments

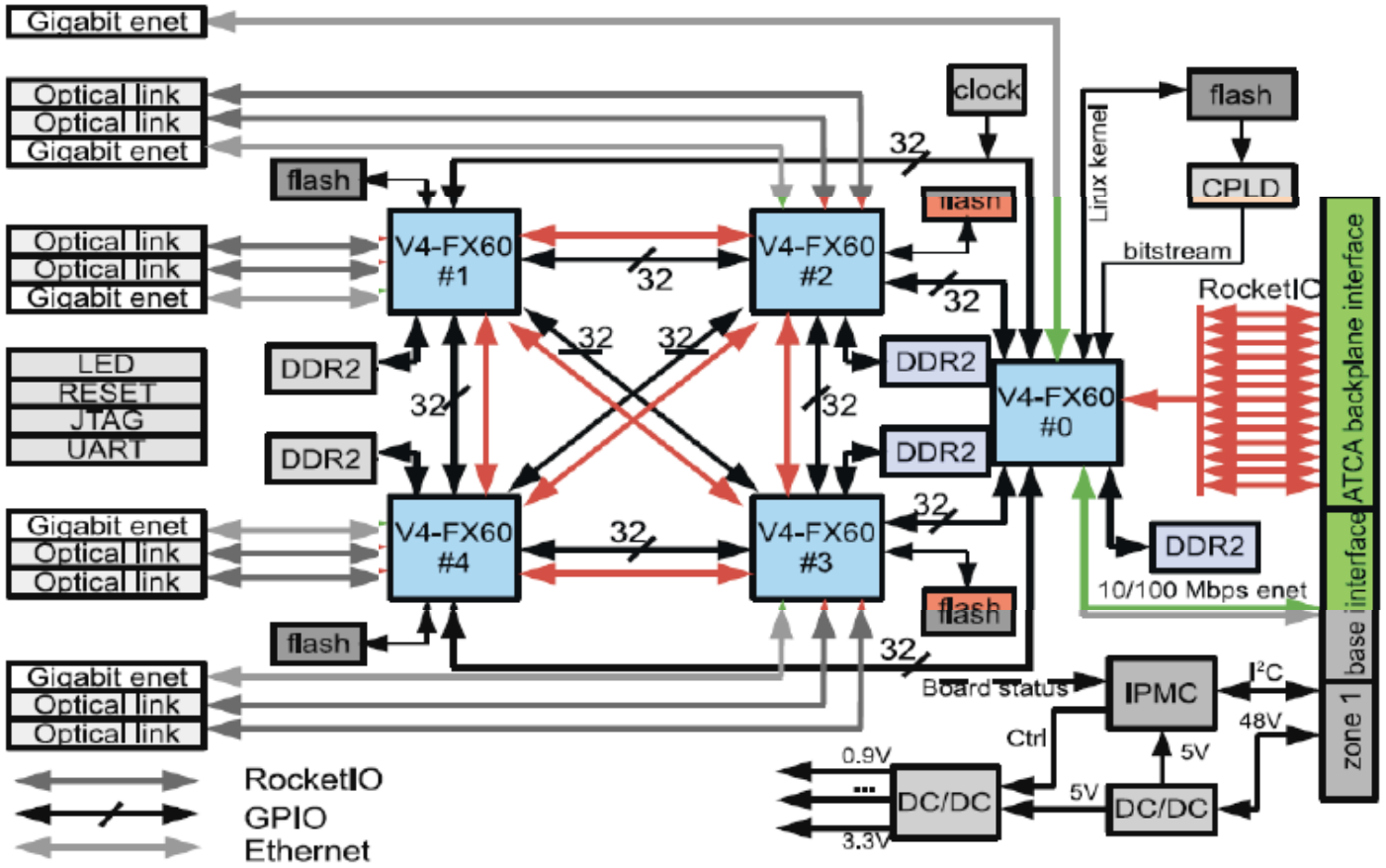
Event Building: ATCA Full Mesh Backplane



point-to-point connections

Event Building: Compute Node Architecture

RocketIO Topology [4 x 4] <> 1 switching FPGA <> 13 (backplane)



HOW TO SYNCHRONIZE DATA FRAGMENTS?

- There are 3 ways
  - a.) 8-bit event number (also in SVD data)
  - b.) 64-bit time stamp (Nakao-san, 19.11.2009)
  - c.) PXD counter synchronized with the grand system clock (beam clock 508.89 MHz, divided by 5 on DHH), written into the data on frontend (Hans Krüger, DAQ MiniWorkshop Giessen 07.08.2009)
- TDR
  - for PXD subevent building, we use the PXD counter to identify event fractions, which belong to each other
  - For PXD+SVD subevent building, we use a lookup table [PXD counter :: event number], synchronize by event number, and write the event number into the data this lookup table needs to be synchronized
  - 64-bit timestamp will not be in the PXD events (unless DAQ group requires it for the global event building)

SVD parasitic readout by PXD system

- Filter algorithm needs PXD and SVD raw data
- HOW MUCH SVD DATA?
- Prior estimate: **(see Ringberg talk, S.L.)**
  SVD raw data = 3 kByte x 30 kHz = 90 MB/s
  = 1 optical link
- New estimate:
  for testing full raw data processing must be assumed
  80 optical links (80 FADCs x 24 APVs)
  each optical link 45 MB/s (~1/6 of PXD raw data)
  this means: **2nd ATCA crate required**

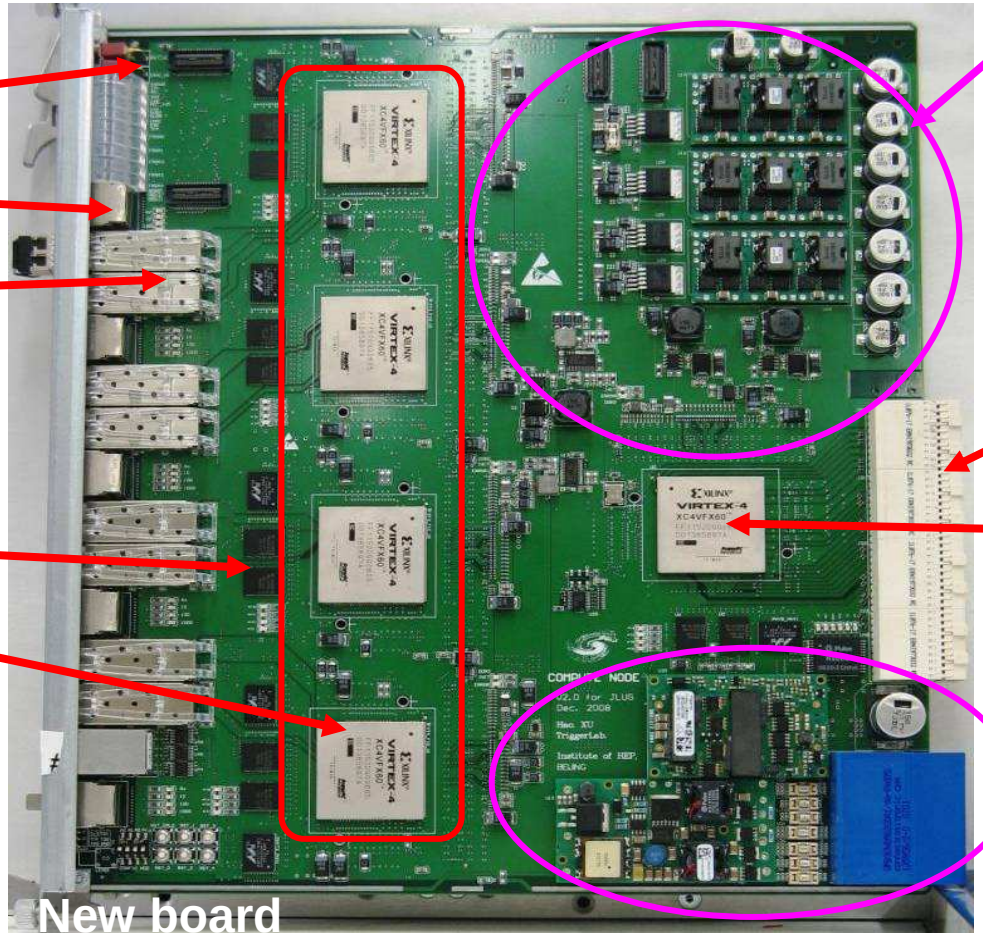
Based on discussion with
Zbigniew Natkaniec and Markus Friedl

SVD parasitic readout by PXD system, cont'd

- Requires 2nd event building in 2nd ATCA
- Challenge:
  How to match
  0.6 Mill. PXD data fragments in one ATCA crate with
  0.6 Mill. SVD data fragments in a 2nd ATCA
  based upon a 8-bit event nr.?
  We need experience from the HADES system.
- Reminder:
  SVD events are discarded after the filter algorithm
  (not sent to EVB)
- Side remark:
  SVD processing takes some time
  pedestal subtraction, 2-pass common mode correction,
  hit finding, hit-time reconstruction
  $\simeq$ 50 us

Status of Compute Node

Compute Node PCB



IPMC

Gbit Ethernet (x5)

Optical link(x8)

FLASH(x10)

Processor

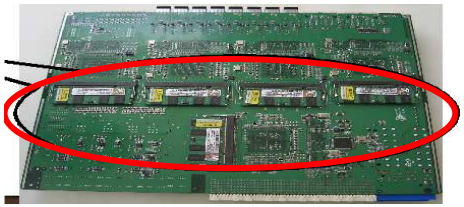
FPGA (x4)

Power Supply

ATCA full mesh neighbor link

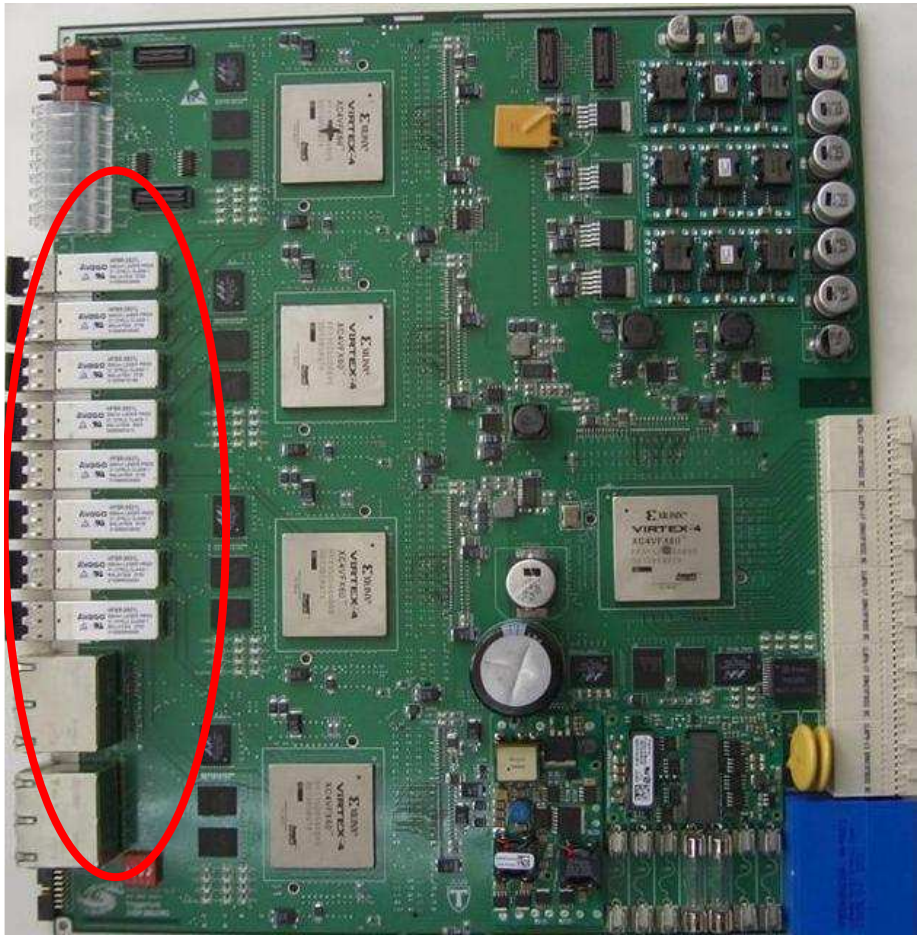
Switch FPGA

Power Supply

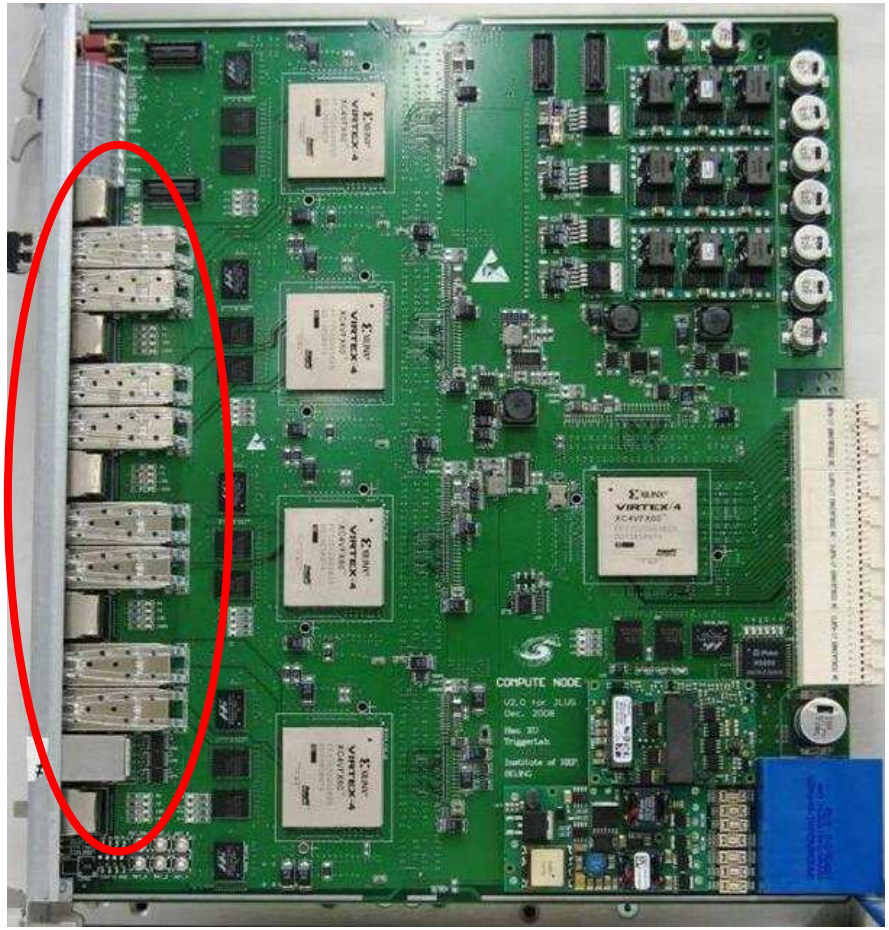


DDR2(backside, x5)

Hardware modification



Compute Node
Version #1, 2008

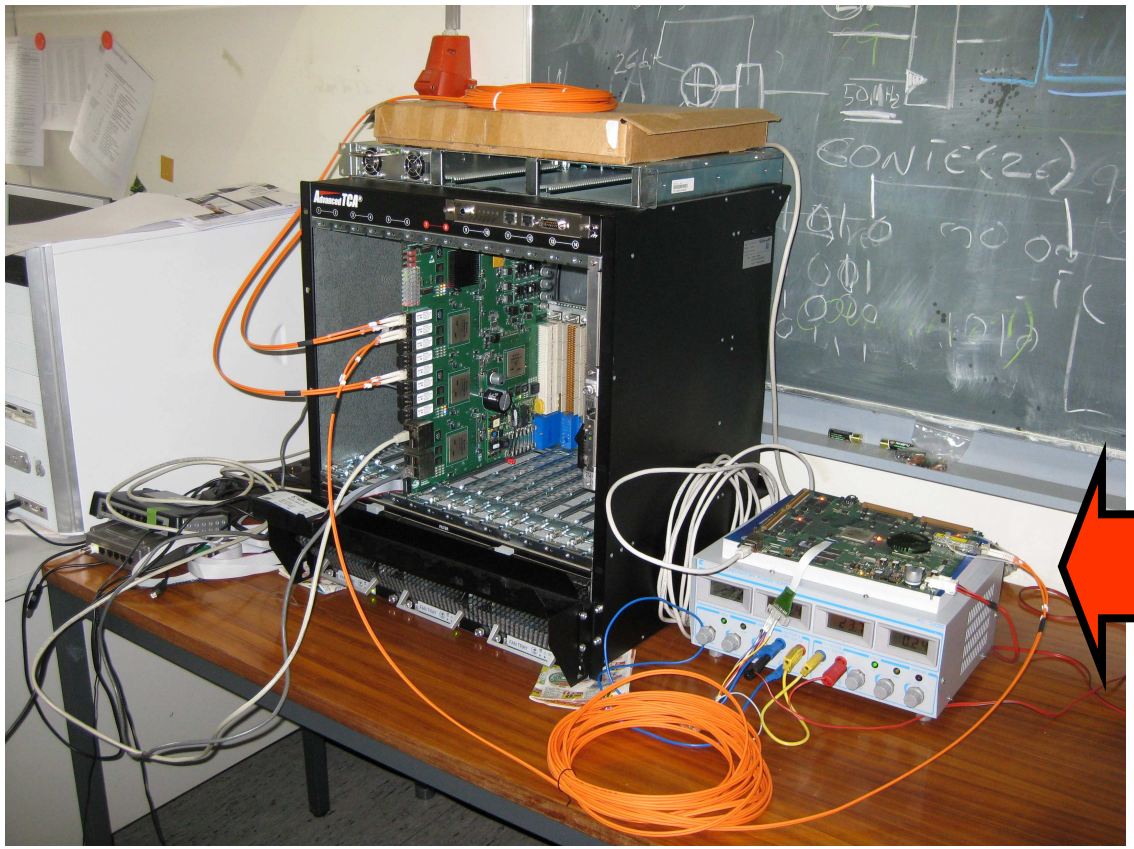


Compute Node
Version #2, 2009

Performance test results with new board

Test #1	data flow, optical link → DDR2 SDRAM
Test #2	P2P transmission on backplane
Test #3	Gb ethernet network performance

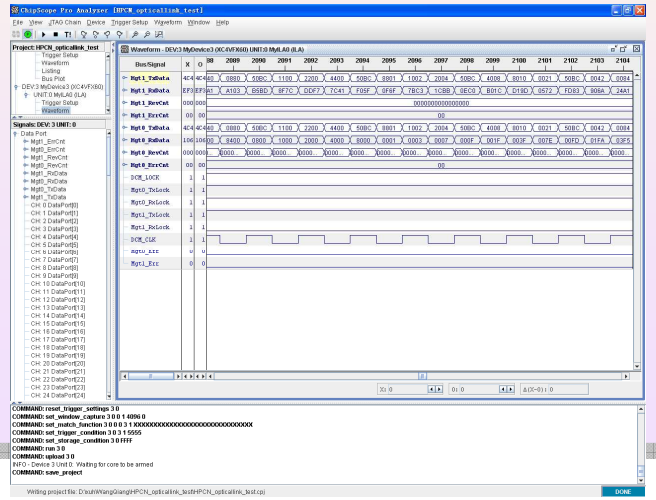
Example system for data flow test



TRB
= HADES
Trigger and
Readout Board
(128 channels,
CERN HPTDC chip)

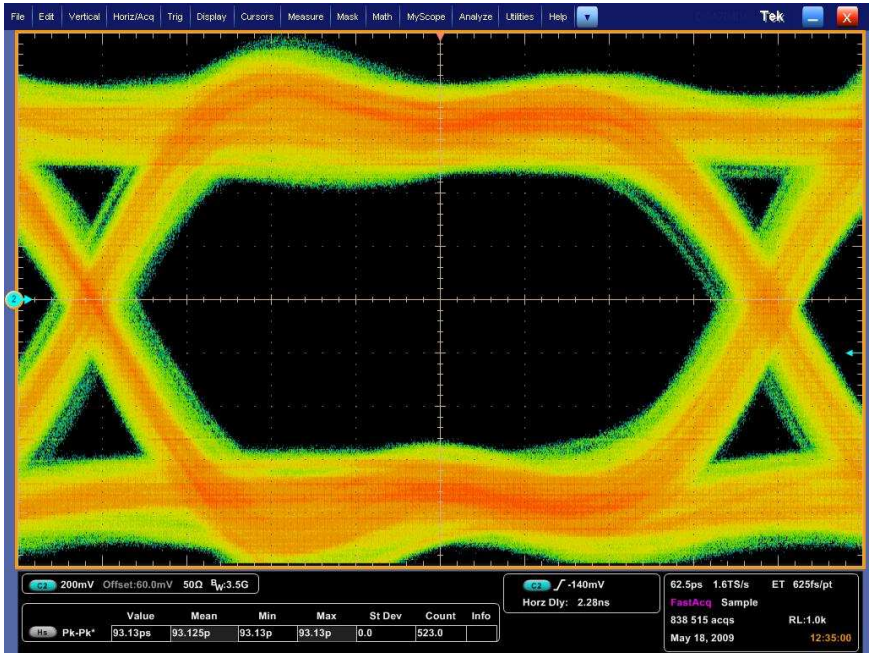
Results: data flow test for 1 readout channel

- optical link → DDR2 RAM → Gb ethernet → PC
  - Generate pseudo-random data on transmitter
  - Receive data from transmitter with optical link
  - Write data into DDR2 RAM
  - Read data from DDR2 RAM and send to PC
  - Check data with software program on PC
  - ~150 hours, ~25 MB/min, 0 error
  - speed limited
    by the checking program
    (chipscope and/or logic analyzer)
  - No data package lost
    at full transmission speed
    (~1.6 Gb/s) of optical link

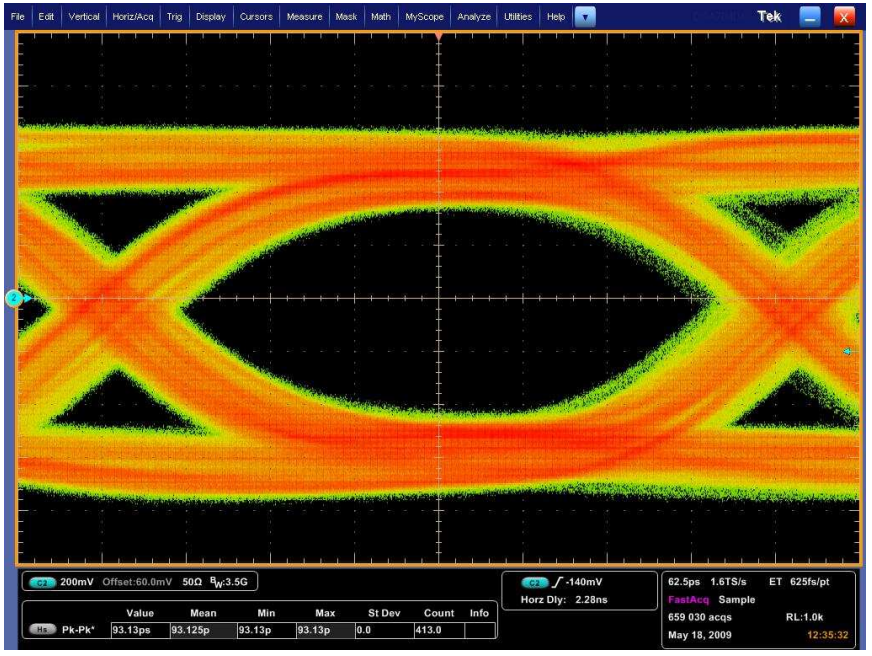


Test #2: P2P Transmission on ATCA Backplane

- 2 CN boards in 1 ATCA shelf, slot #1 and slot #14
  i.e. the longest line in full mesh backplane
- eye-diagram: small jitter and small overshoots



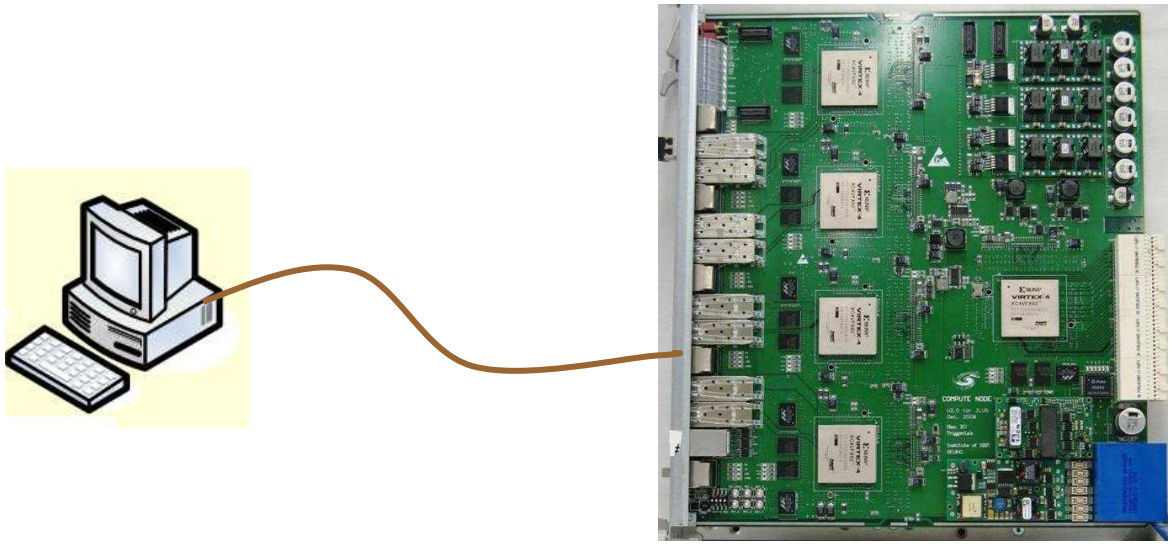
Transmitter



Receiver

Test #3: GB Ethernet Network Performance

- TCP/IP protocol
- Linux (Kernel 2.6) or Vxworks 5.5 on PowerPC405 (as hardcore on Virtex-4 FPGA)



Linux on PowerPC405 on FPGA

```
Linux/PowerPC load: root=/dev/nfs ip=192.168.0.4:192.168.0.3:192.168.0.3:255.255.255.0 rw nfsroot=192.168.0.
home/mingliu/ml403_rootfs console=ttyUL0,9600 mem=64M mtdparts=physmap-flash.0:6M(kernel),42M(others),-(bits
am)
Finalizing device tree... flat tree at 0x40ae18
Using Xilinx Virtex machine description
Linux version 2.6.27-rc9 (mingliu@cca03) (gcc version 3.4.1) #15 PREEMPT Thu Nov 27 11:57:06 CET 2008
Zone PFN ranges:
  DMA      0x00000000 -> 0x00004000
  Normal   0x00004000 -> 0x00004000
  HighMem  0x00004000 -> 0x00004000

ML300 powerpc linux 2.4.21-pre7 E.I.S. edition
192.168.0.4 login: root


Welcome to the ML300, EIS edition

Be careful, it's blue.

# cd /home
```

Network Performance – Linux

PowerPC405->PC

```
# ./netperf -l 10 -H 192.168.0.1 -c -C -t TCP_STREAM -- -m 65536 -s 253952 -S 253952
TCP STREAM TEST from 0.0.0.0 (0.0.0.0) port 0 AF_INET to 192.168.0.1 (192.168.0.1) port 0 AF_INET
```

Recv Socket Size bytes	Send Socket Size bytes	Send Message Size bytes	Elapsed Time secs.	Throughput 10^6bits/s	Utilization Send local % U	Utilization Recv remote % T	Service Demand Send local us/KB	Service Demand Recv remote us/KB
253952	217088	65536	10.01	**212.37**	-1.00	5.01	0.000	1.932

PC->PowerPC405

```
$ ./netperf -l 10 -H 192.168.0.4 -c -C -t TCP_STREAM --  -m 65536  -s 253952 -S
 253952
TCP STREAM TEST to 192.168.0.4
```

Recv Socket Size bytes	Send Socket Size bytes	Send Message Size bytes	Elapsed Time secs.	Throughput 10^6bits/s	Utilization Send local % T	Utilization Recv remote % U	Service Demand Send local us/KB	Service Demand Recv remote us/KB
217088	253952	65536	10.01	**241.42**	1.57	-1.00	0.532	-0.339

Vxworks 5.5 on PowerPC405 on FPGA

```
ComputeNode vxworks start:

Executing program starting at addrerTarget Name: vxTarget
lltemac: warning - MII clock is defaulted to 1000 Mbps
lltemac: Buffer information:
        0x00018810 bytes allocated for all buffers
         1564 byte cluster size
           60 Rx buffers, 1st buffer located at 0x07E396C0
            4 Tx buffers, 1st buffer located at 0x07E50640
        0x00001008 bytes allocated for all BDs
           32 RxBDs, BD space begins at 0x07E36900
           32 TxBDs, BD space begins at 0x07E37100
Attached TCP/IP interface to lltemac unit 0
Attaching network interface lo0... done.


NETWORK PERFORMANCE TEST SUITE
Created on May  9 2009

PPC405 core          : 300 MHz
Stack                : VxWorks

Initializing...
Frame buffers at     : heap space
Frame buffer bytes   : 0x00600000
Descriptors at       : 0x07FE0000
Descriptor bytes     : 0x00020000
```

Network Performance – Vxworks 5.5

PowerPC405->PC

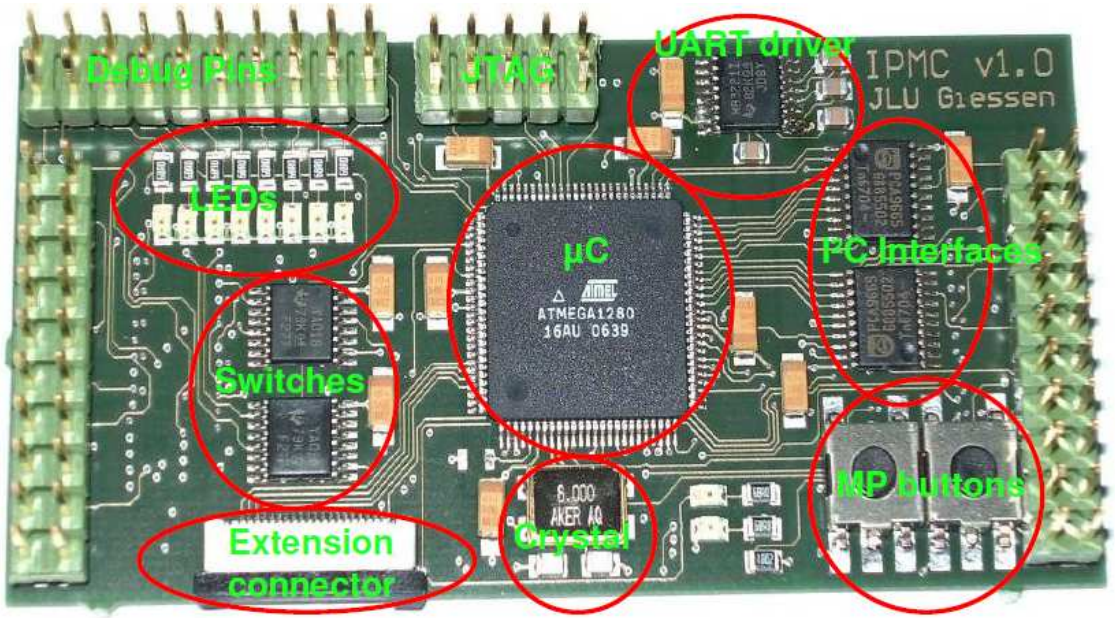
```
Main Menu:
1  - Set New Transport Characteristics
2  - Show Transport Characteristics
3  - Set Link Speed (1000 FD)
4  - Set Netperf task priority
5  - Tx UDP Stream
6  - Tx TCP Stream
7  - Tx Canned UDP & TCP menu
8  - Rx Start Netperf Server (stopped)
9  - Util menu
100- Exit
Enter selection: 6
Size in bytes of local socket buf [253952]:
Size in bytes of remote socket buf [253952]:
Size in bytes of message [65536]:
Number of seconds to run test [5]: 300

TCP STREAM TEST to 192.168.0.4
Recv    Send    Send                          Utilization       Service Demand
Socket Socket  Message  Elapsed              Send     Recv     Send    Recv
Size   Size    Size     Time     Throughput  local    remote   local   remote
bytes  bytes   bytes    secs.    10^6bits/s  % U      % U      us/KB   us/KB

253952 253952  65536    300.00     210.19    100.00     -1.00    0.000   -1.000


Main Menu:
1  - Set New Transport Characteristics
2  - Show Transport Characteristics
3  - Set Link Speed (1000 FD)
4  - Set Netperf task priority
```

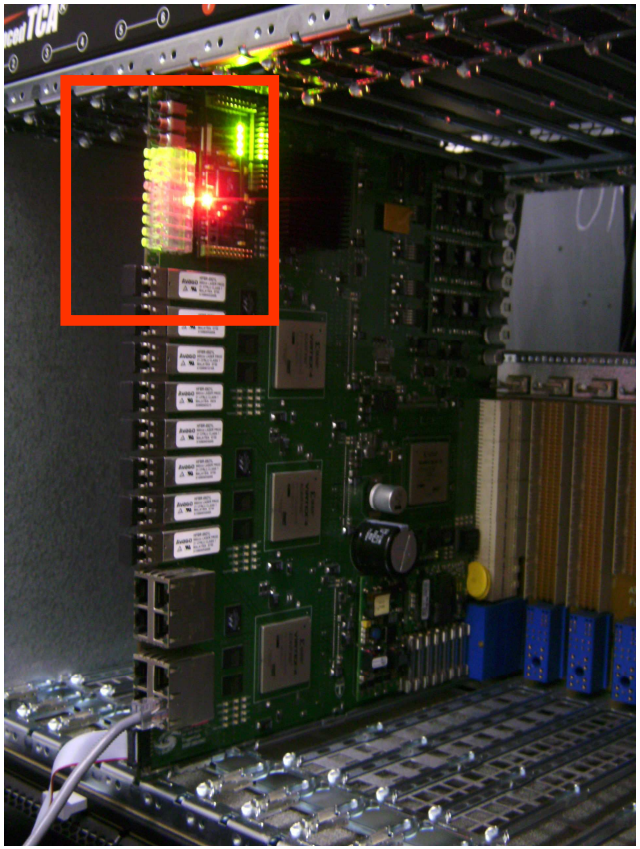
Implications of the Test Result

- 200 Mb/s = 25 MB/s
- Reminder:
  at 1% occupancy, f_1=10, f_2 x f_3=10
  requirement is 30 MB/s
- but this test result is PowerPC,
  limited by speed of CPU (300 MHz) and
  interrupt handler (for each 9.6 kB packet)
- Plan:
  send data by FPGA instead of PowerPC
  expected to be faster
  results from XILINX indicate $\geq$ 100 MB/s
  requires implementation of TCP/IP stack
  (project for $\geq$ 1 year)

ATCA Management: IPMI Controller Add-On Board

Intelligent Platform Management Interface

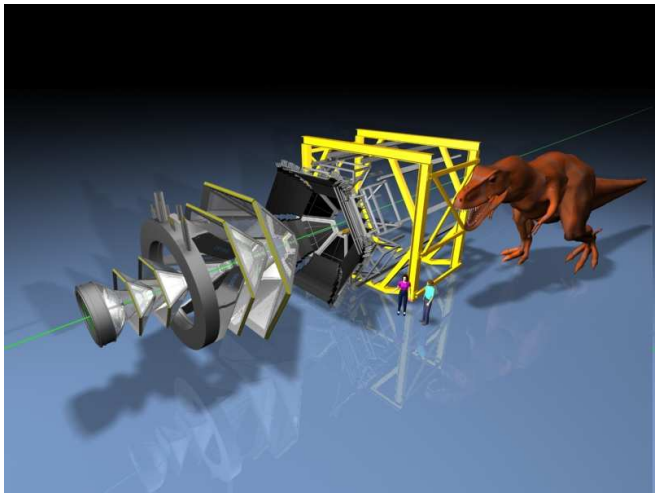




Problem with UART solved
2nd PCB soldered 2 weeks ago
under testing

A Planned Full System Prototype for HADES

- 1 full ATCA system with 12 CN
- 20 kHz L1 rate
- 377 MB/s in spill
- 60k detector channels
- 73 optical links, 1.6 Gb/s
- TRB
  HADES Trigger and Readout Board
  IEEE Trans. Nucl. Sci. 55(2008)59
- Algorithms:
  - event building
  - event decoding
  - drift chamber track finder
  - RICH ring finder
  - matching with TOF and SHOWER



What is needed in addition to the HADES system?

- Proposed HADES and PXD systems are identical
- Exception:
  2 hardware modifications
  of the Compute Node,
  only required by PXD DAQ

1. 4 GB memory per FPGA
   more buffer for waiting
   for tracks from L3 farm
   > impossible
   requires a complete new
   compute node PCB

2. optical links 6.5 Gb/s
   note: SFP+ transceiver has 8 Gb/s,
   but limit is RocketIO with 6.5 Gb/s


SFP


SFP+

No manpower
No funding

TODO, large scale projects (≥ 1 year)

- Hardware
  - Trigger distribution system
  - Timing synchronization system
  - DHH (Göttingen)
  - 2nd ATCA system for SVD readout
- Firmware (VHDL):
  - Event building PXD
  - Event building SVD (input for filter algorithm)
  - Synchronisation PXD-SVD

Summary

- There is a substantial amount of work.
- There are several non-covered subprojects
  of considerable size
  (e.g. hardware trigger).
- We have to define interfaces to SVD.
- There are groups interested in support
  IHEP
  and some other groups.
- General timescale is shifted by 3 years:
  PXD DAQ construction phase 2012-2014 (if any support)
  PXD DAQ debugging and testing 2014-2015 (if any support)