The ATCA System – Tasks and Testbenches

Björn Spruck for the Giessen Group
II. Physikalische Institut, JLU Gießen
11.05.2011
7th International Workshop on
DEPFET Detectors and Applications, Ringberg

Outline

- Challenge
- Implementation
- Tests and benchmarks
- Summary and outlook

II. Physikalisches Institut
JUSTUS-LIEBIG-UNIVERSITÄT GIESSEN

bjoern.spruck@physik.uni-giessen.de

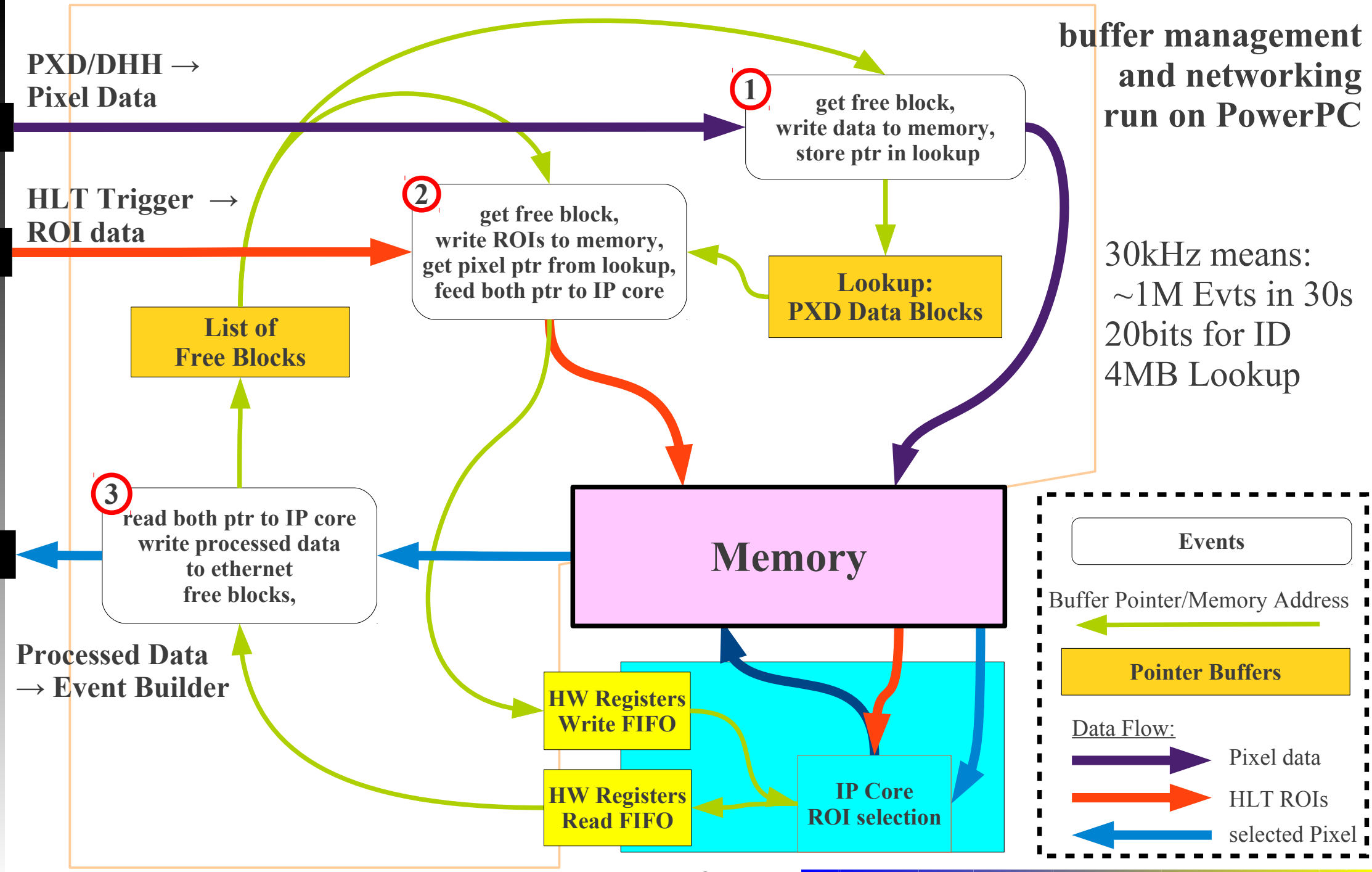
Belle II

Challenge

- Task 1
  - Buffer data until High Level Trigger decision
  - **Rate reduction** by 1/5
  - HLTs are in random order and have up to 5s latency!
  - Dynamic management of memory buffers
- Task 2
  - Select Pixel data inside regions of interest (ROIs)
  - ROIs from HLT and SVD tracklet
  - **Size reduction** 1/10 per accepted event

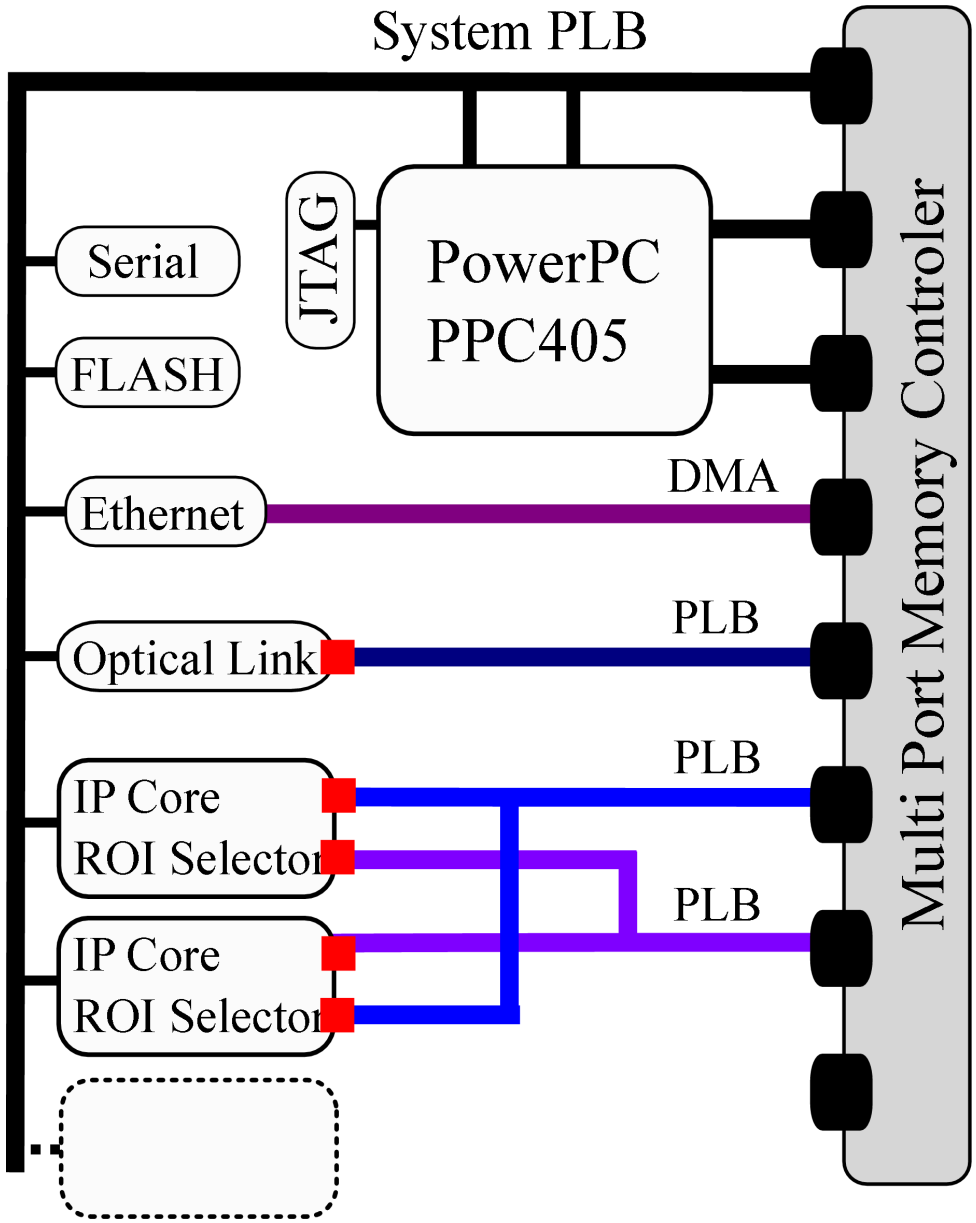
- We assume:
  - 4 bytes per pixel (worst case), 32 bit aligned
  - ~550MB/s input, 30kHz event rate

System Layout – Event/Software View

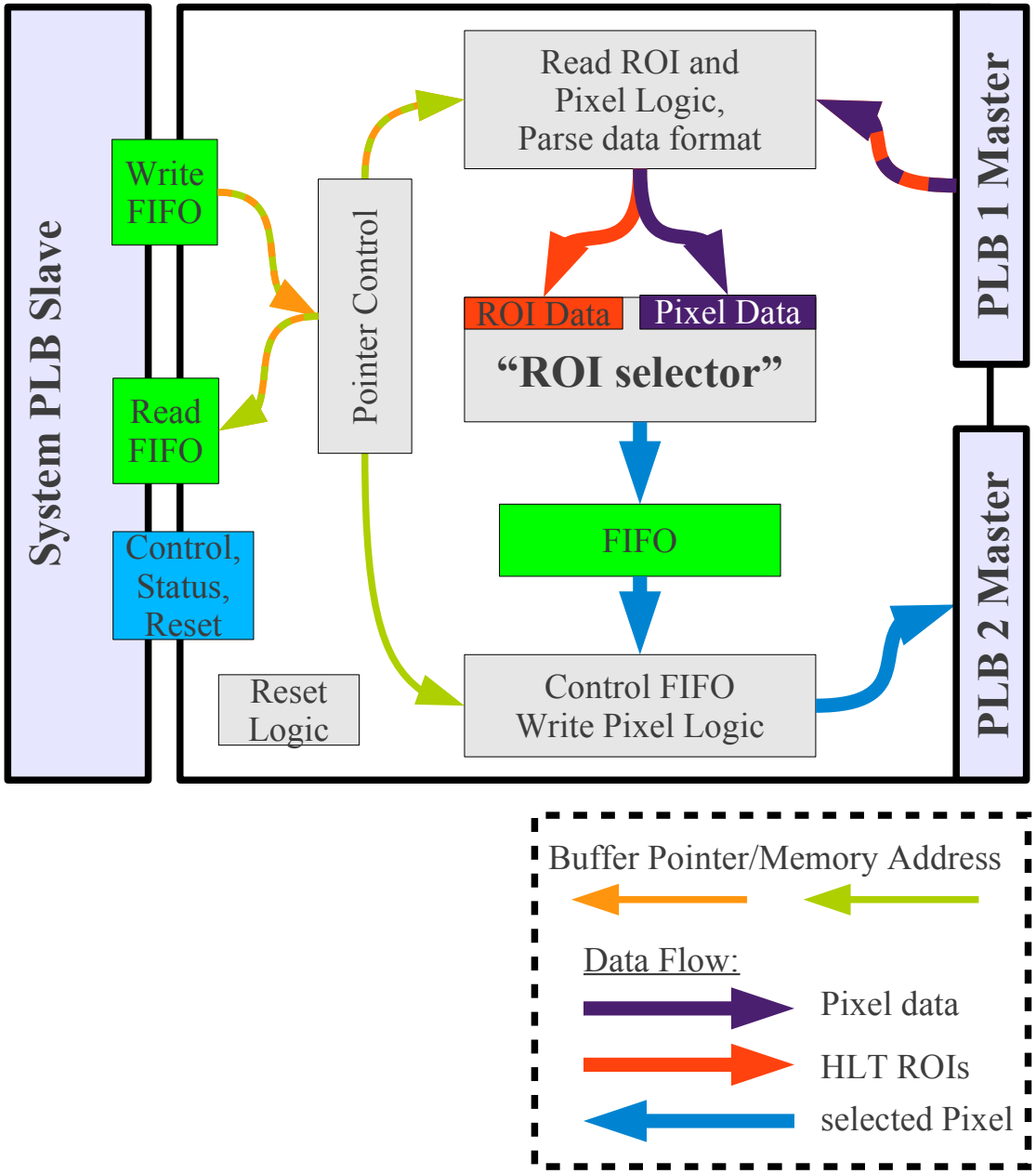


System Layout

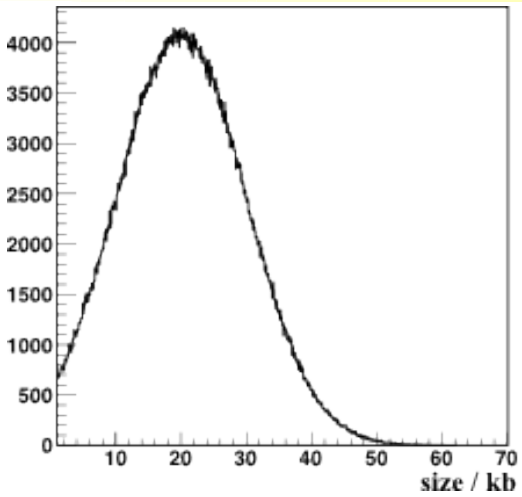
FPGA - Virtex 4 FX 60

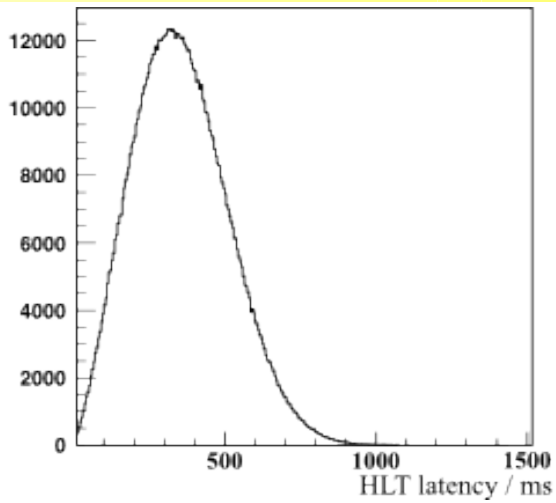


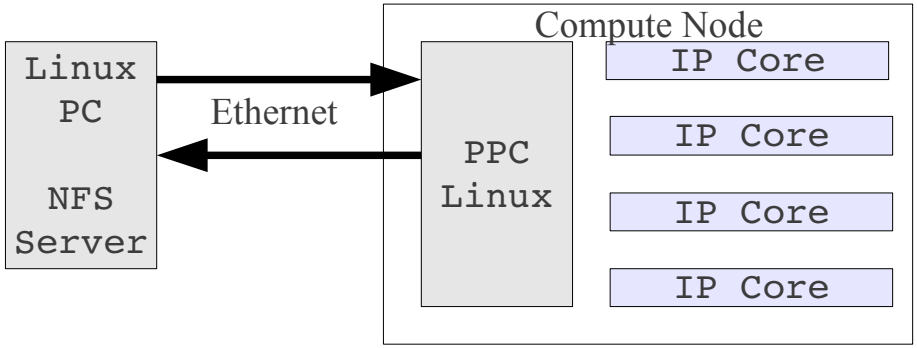
IP Core – ROI Selector



Testbench





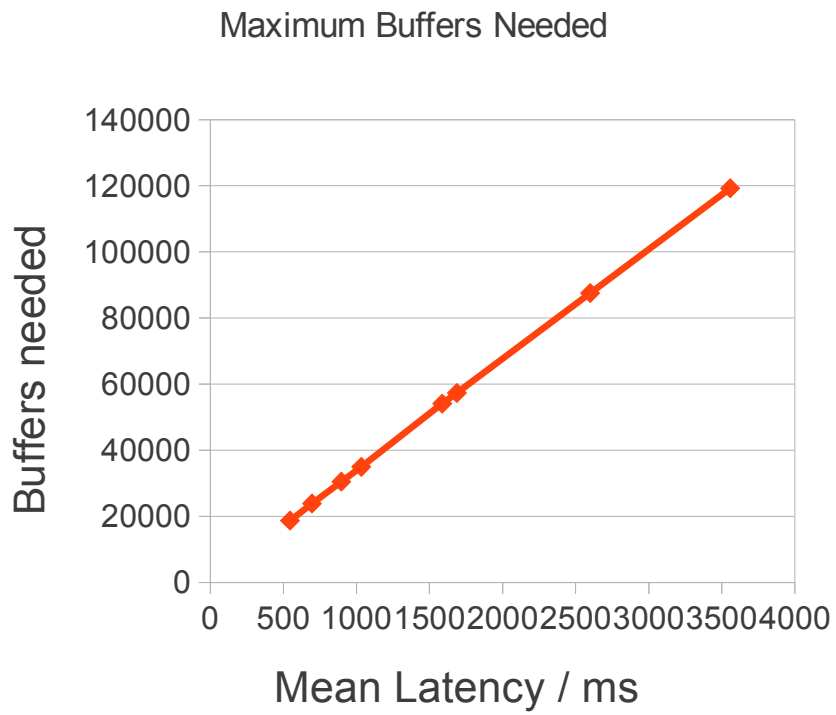


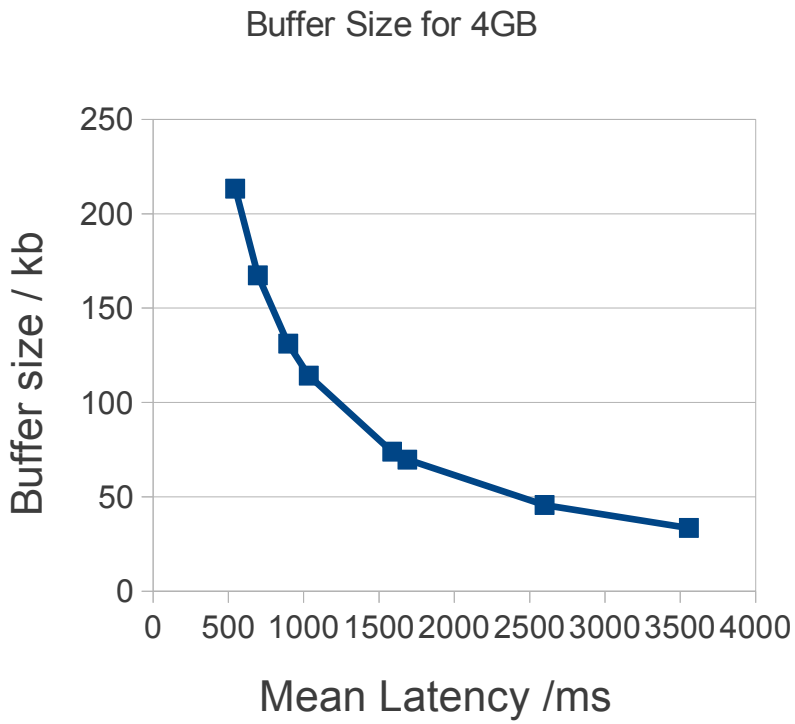
(no optical link used up to now)

- Test with random data and ROIs
- The pixels are either rejected <u>or</u> marked by the IP core (debugging)
- Setup 1:
  - Generate and check data on CN, compare to IP core result
  - ~400k evts with 10k pixel and 31 ROIs per IP core → no error in $40 \cdot 10^9$ pixel
- Setup 2:
  - Generating and Checking on PC, CN is only processing data
  - TCP/IP server/client programs
  - 3MB/s in and out, limited by TCP/IP stack on PPC (or 5MB/s in, 1MB/s out)

Test 1: Memory/Buffer consumption

Result is maximum buffer number used
=> Buffer size depends on Memory and Latency
For fixed size: Memory usage scales with buffer size.





Test 2: Throughput

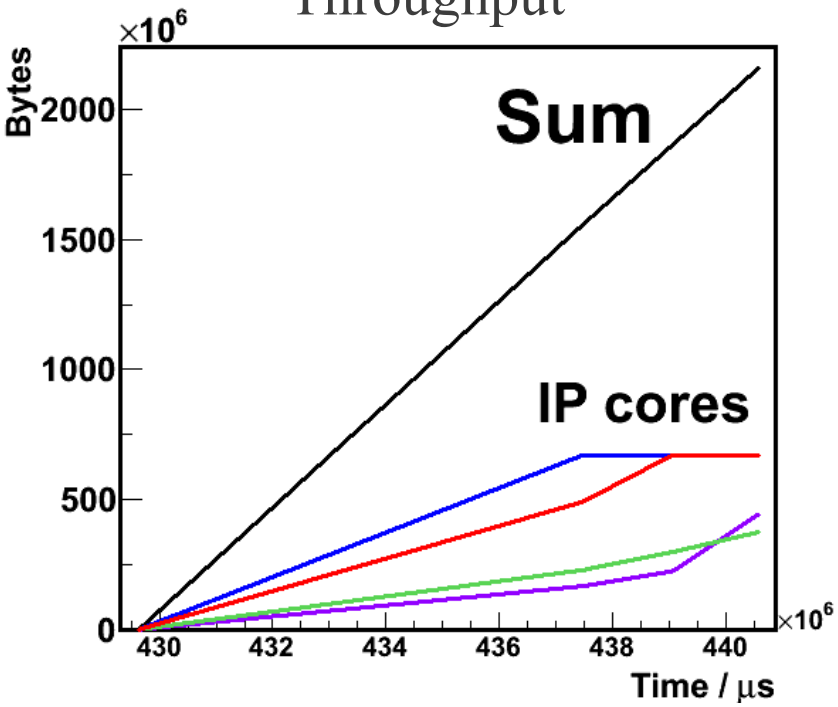
- Memory bandwidth test:

Type	Cores	PLBs	Throughput	Sum
Plain Fifo Copy	1	1	100 MB/s	200 MB/s
Plain Fifo Copy	2	2	180 MB/s	360 MB/s
Full Core, 31 ROIs	1	2	182 MB/s	364 MB/s
Full Core, 31 ROIs	4	3	198 MB/s	396 MB/s

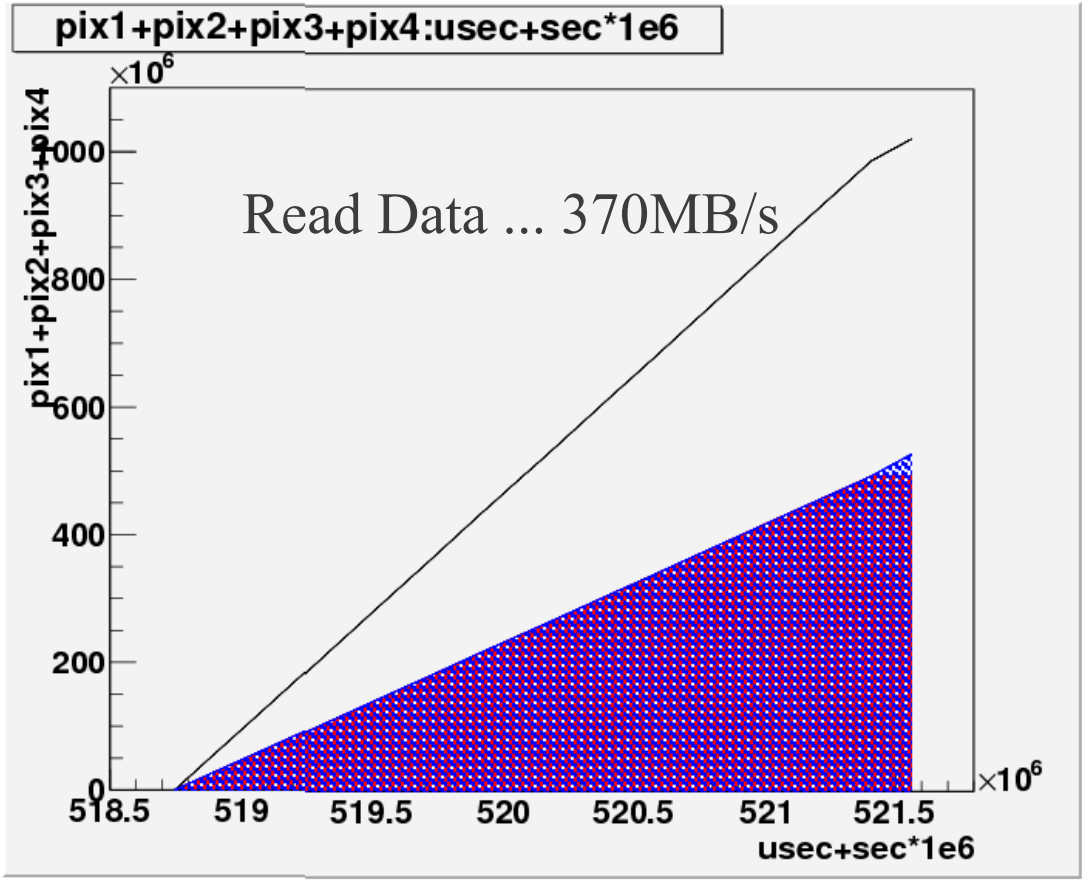
Problem under investigation!

- Throughput of 180MB/s ≙ 1% occupancy
- Limited by memory controller (MPMC)?
  - Wrong parameters/clocks?
- <u>No errors observed!</u>
- Device usage (incl. debugging logic):
  - For a 10 ROI logic 13% Slices
  - For a 32 ROI logic 19% Slices
  - MPMC 32% Slices

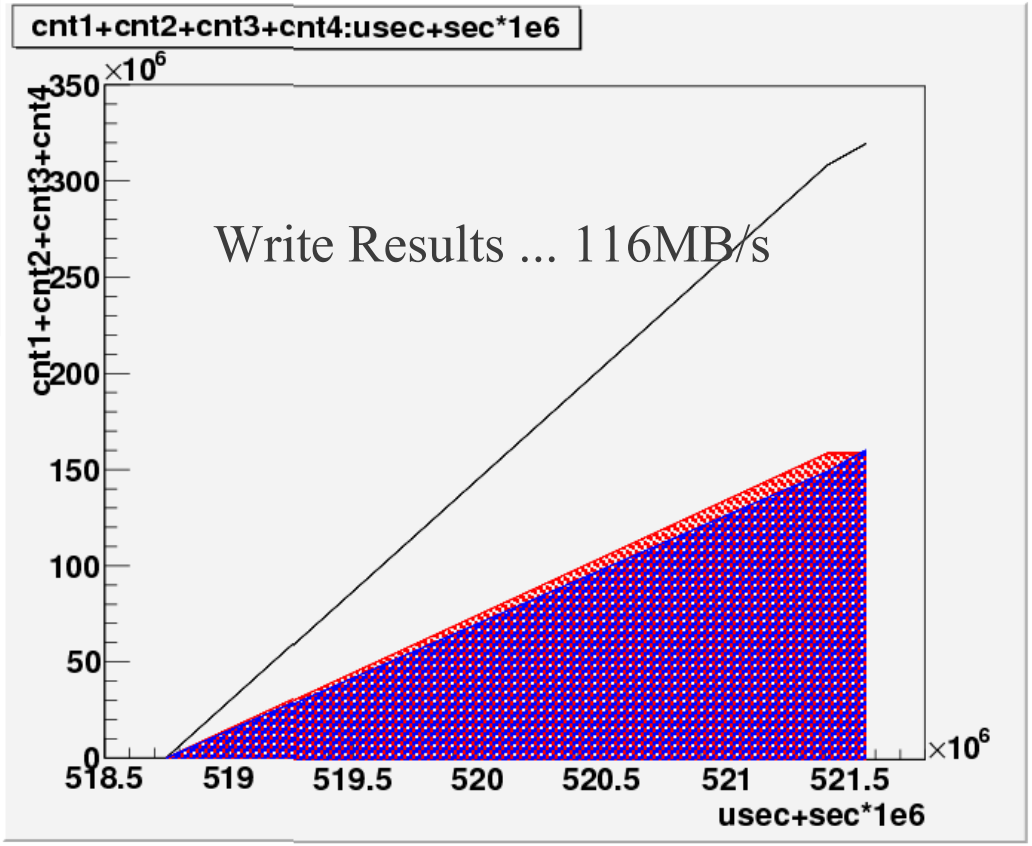
Throughput



Test 3: Throughput with Selection on Data



(two IP cores, MPMC reconfigured)



Note: 550MB/s in, HLT triggers 1/5
→ ~110MB/s read throughput "needed"

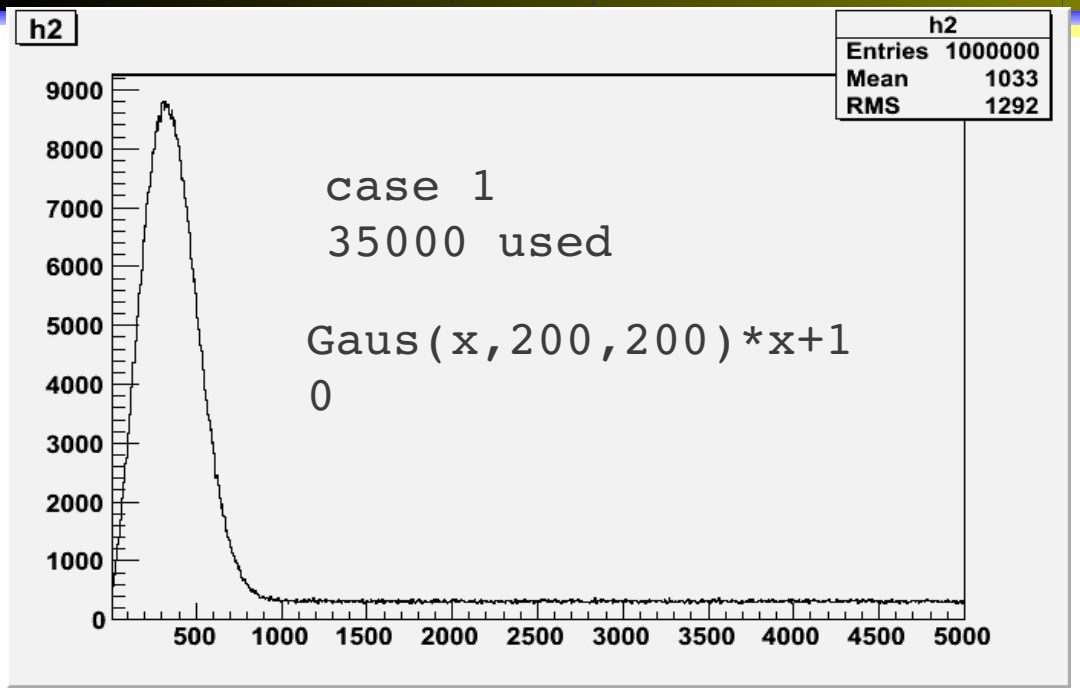
Summary and Outlook

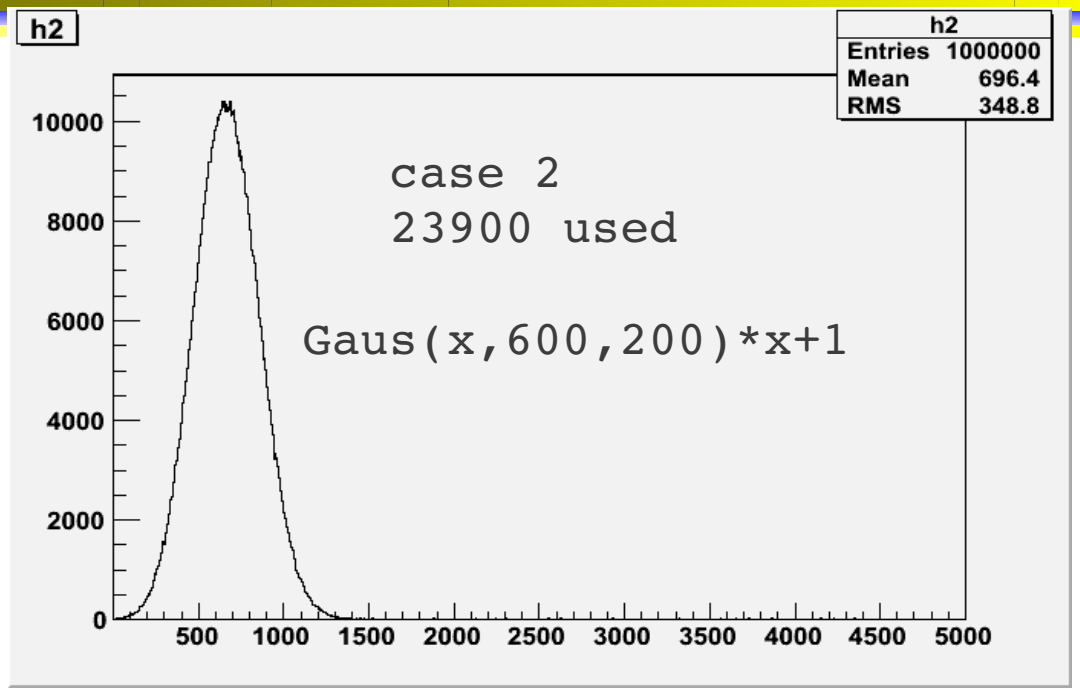
- **It works!** (tested with up to 4 cores on one FPGA)
- Implementation
  - Time critical parts in IP core
  - Buffer handling and TCP/IP data transfer still on PPC
- TODO:
  - Optical Link for data transfer
  - Memory throughput (MPMC?) needed for optical link
  - Use MC data from digitizer level
- Known deviations from optimum
  - PPC Cache "incoherence", access on same memory from PPC and IP core
  - 64 bit Bus → 32 bit data (only every second pixel processed)
    - → 32 bit Bus or higher clock for inner IP core
    - Depacking bits in bytes; data format dependent!
  - Soft reset behavior, pointers returned too early

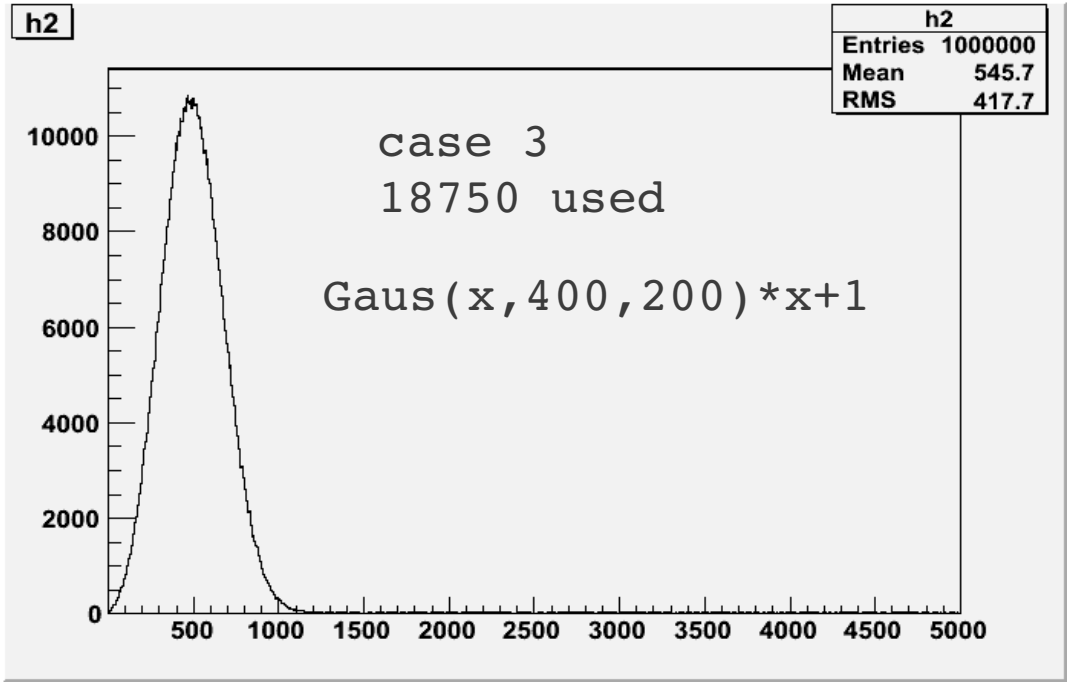
Backups

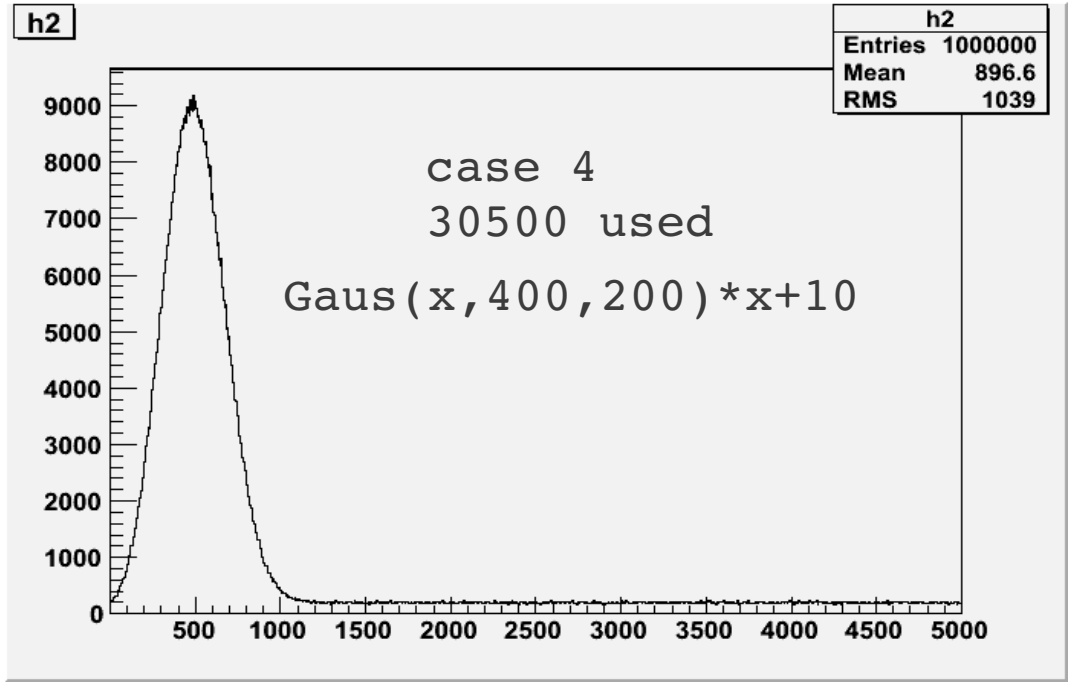
Environment

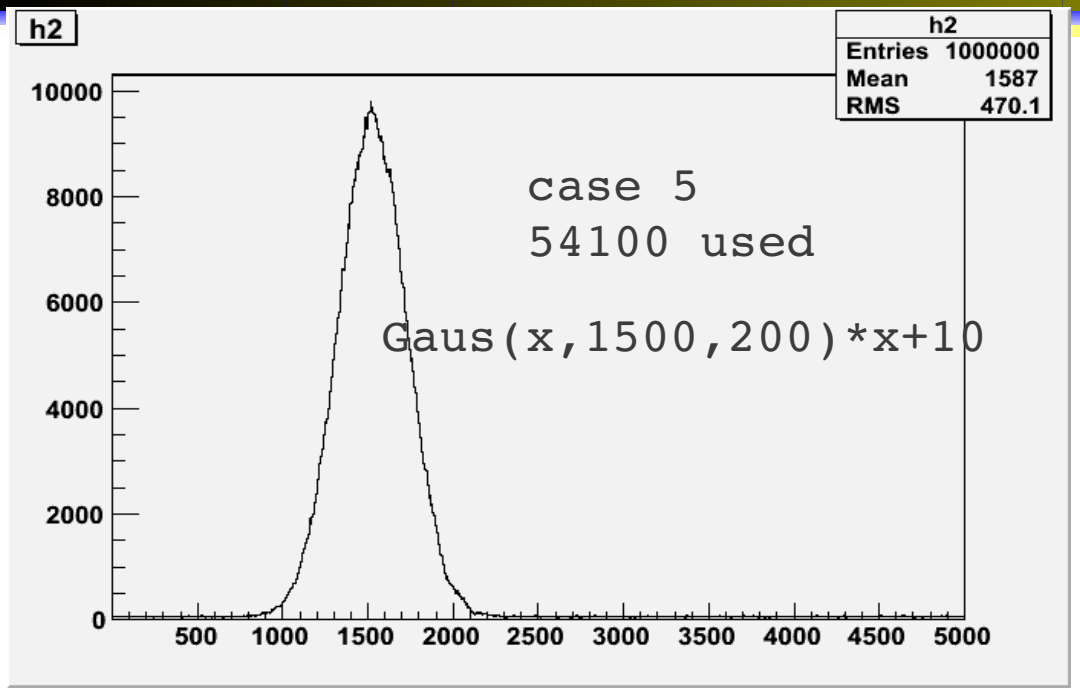
- Operating System:
  - PowerPC embedded linux, kernel 2.6.34 (from 2010)
  - Two setups
    - Standalone from initramfs
    - Complete system by NFS if available; including gcc, ssh, ...
  - “Jumbo” frame support
- Full memory is <u>accessible</u> by PowerPC, but <u>not controlled</u> by Linux
  - Full memory <u>and</u> hardware addresses (IP core) accessable from user program (mapping with “mmap”)
  - No linux kernel driver needed (yet?), <u>not</u> interrupt driven
- Memory access from PowerPC and IP Core → cache problem!
  - Use hardware registers for control
- PowerPC controls the IP cores, but does no computing itself.

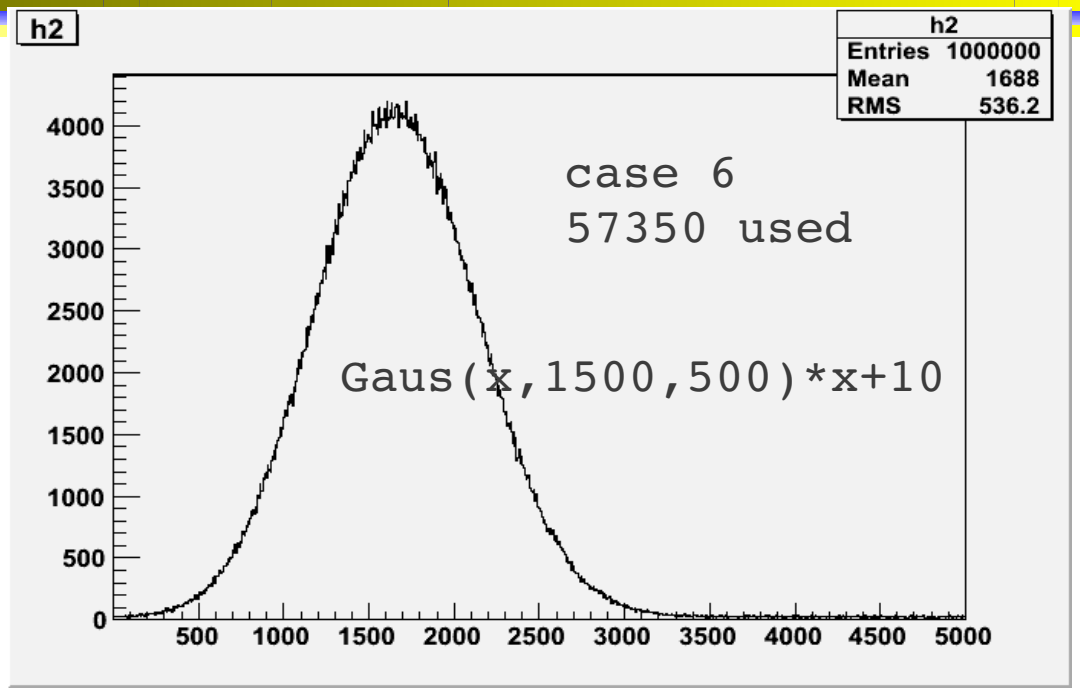

h2
h2
Entries 1000000
Mean 1033
RMS 1292
case 1
35000 used
Gaus(x,200,200)*x+1
0

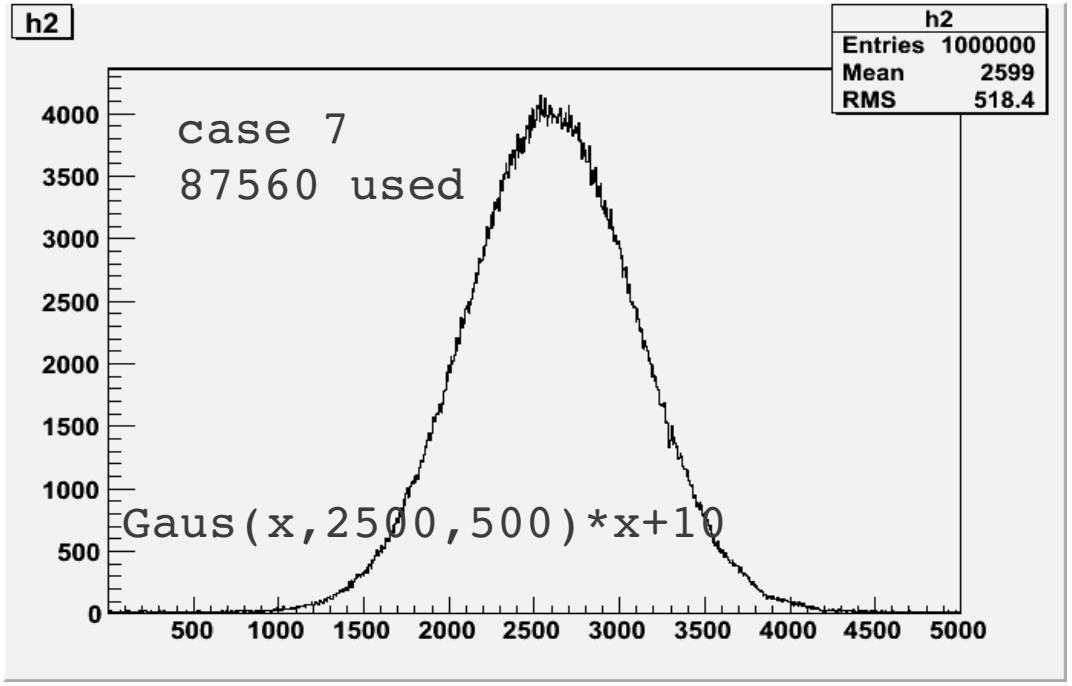

h2
h2
Entries 1000000
Mean 696.4
RMS 348.8
case 2
23900 used
Gaus(x,600,200)*x+1

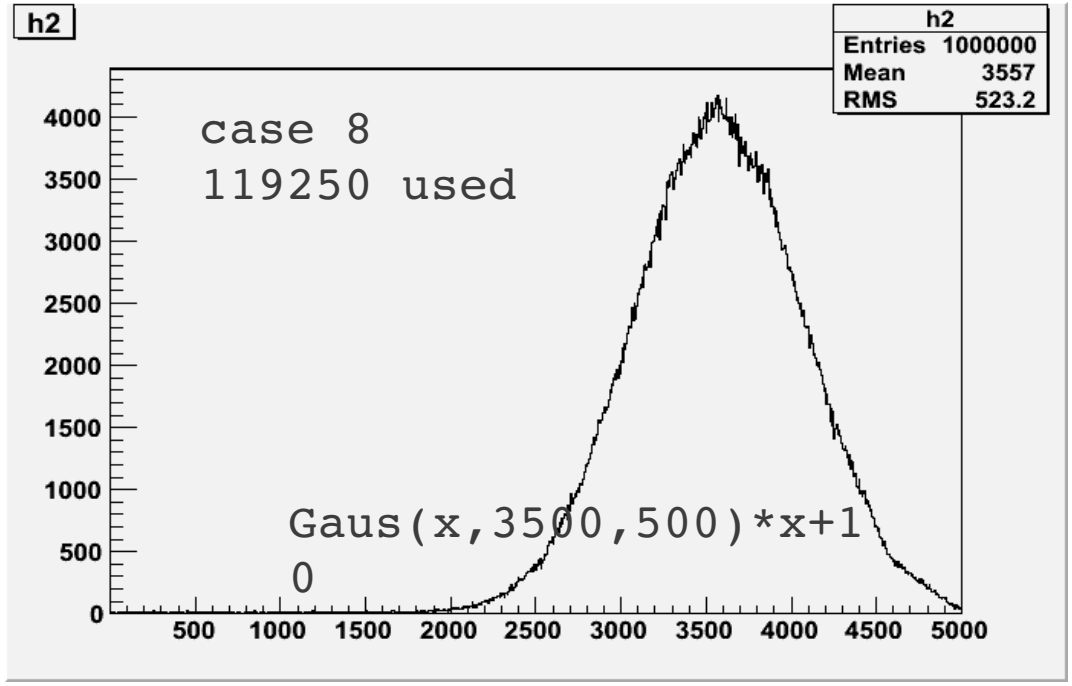

h2
h2
Entries 1000000
Mean 545.7
RMS 417.7
case 3
18750 used
Gaus(x,400,200)*x+1


h2
h2
Entries 1000000
Mean 896.6
RMS 1039
case 4
30500 used
Gaus(x,400,200)*x+10


h2
h2
Entries 1000000
Mean 1587
RMS 470.1
case 5
54100 used
Gaus(x,1500,200)*x+10

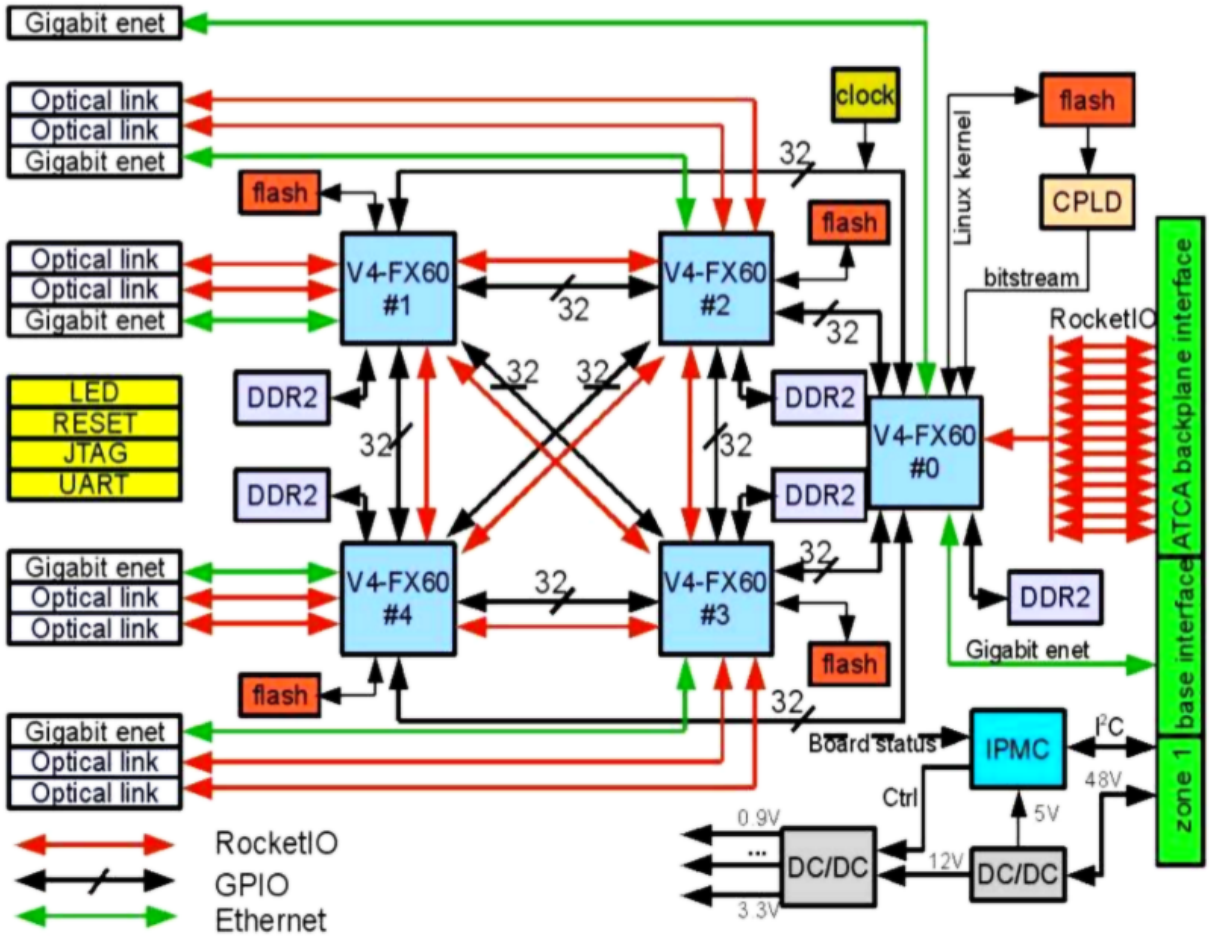

h2
h2
Entries 1000000
Mean 1688
RMS 536.2
case 6
57350 used
Gaus(x,1500,500)*x+10


h2
h2
Entries 1000000
Mean 2599
RMS 518.4
case 7
87560 used
Gaus(x,2500,500)*x+10

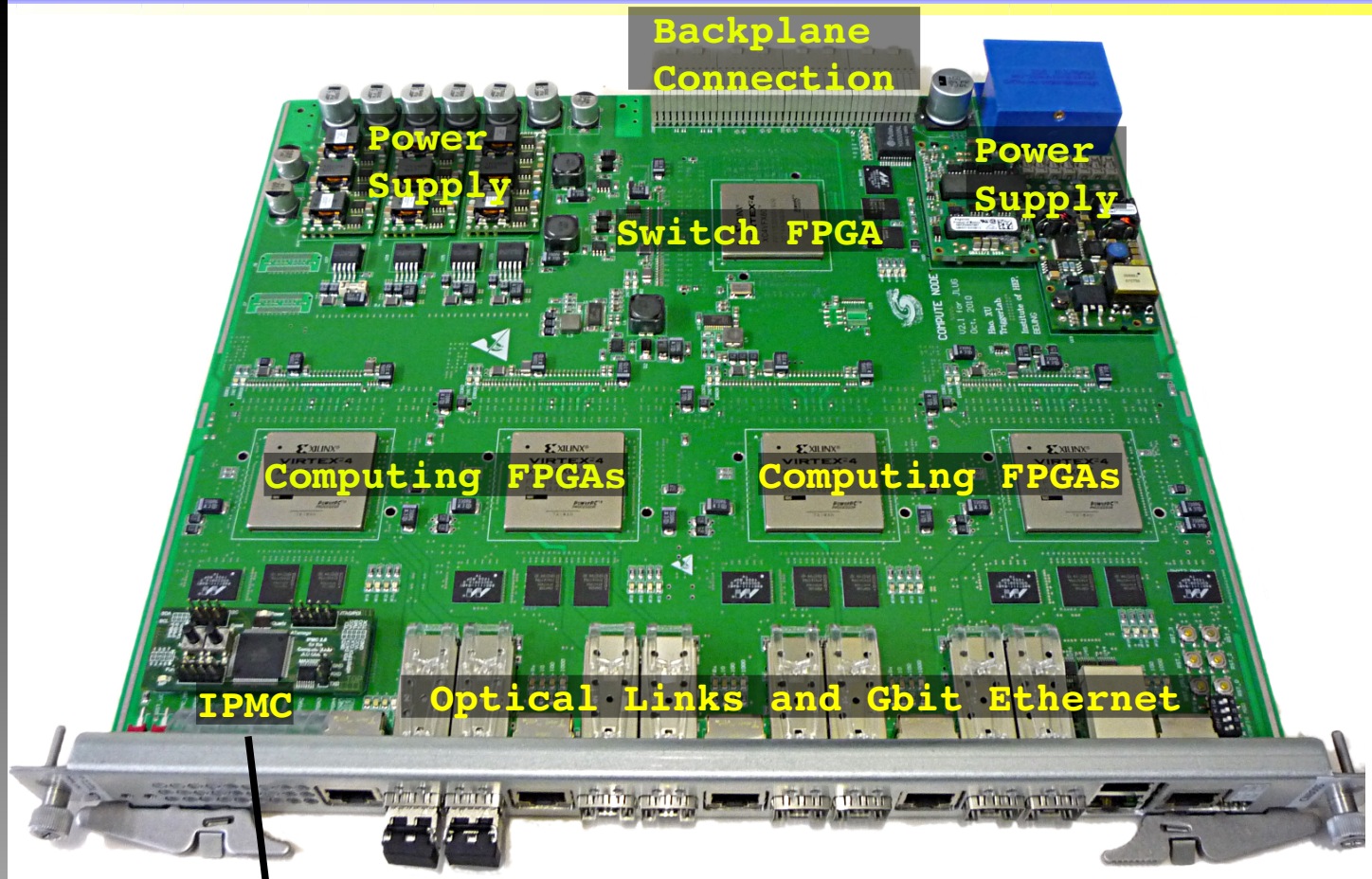

h2
h2
Entries 1000000
Mean 3557
RMS 523.2
case 8
119250 used
Gaus(x,3500,500)*x+1
0

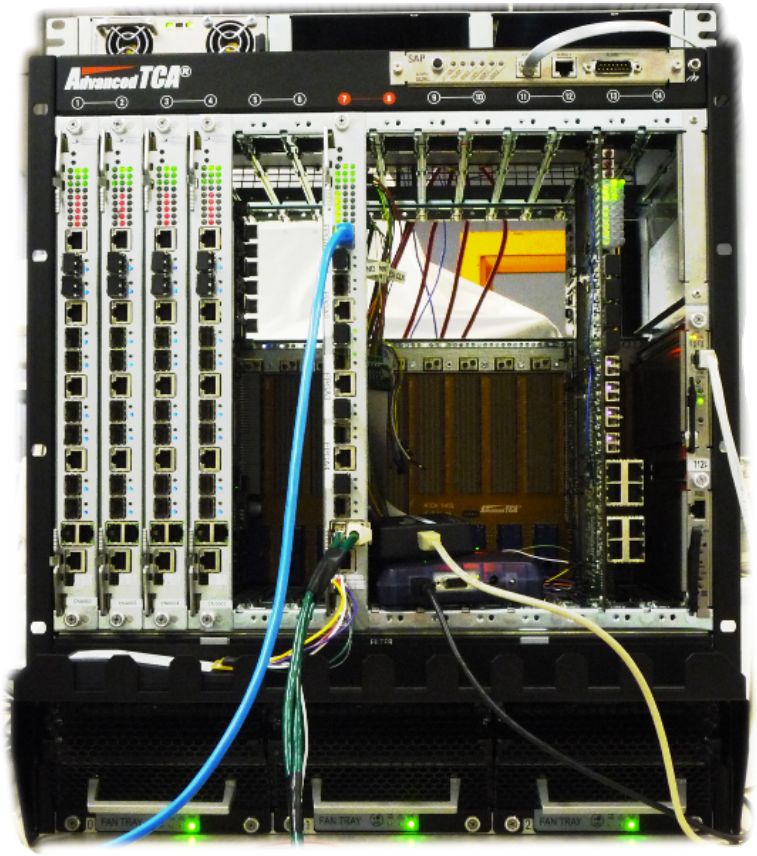
The Compute Node (CN)

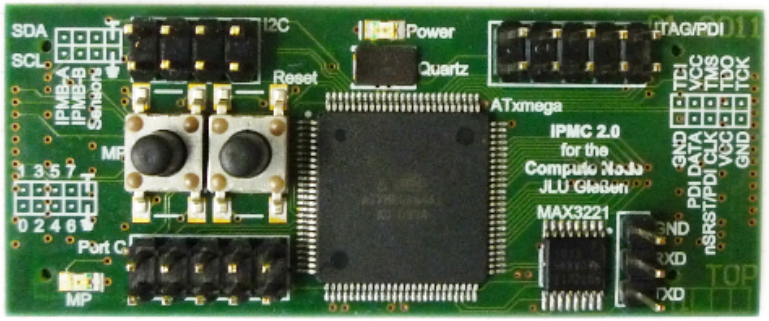
- Developed in close cooperation with Trigger Lab of IHEP Beijing, China
- High Performance Computing
  - 5 Virtex-4 FX60 FPGA
    - (upgrade: Virtex 5)
  - 5*2GB DDR2 RAM
    - (upgrade: 4GB)
  - interconnected by RocketIO
- ~32Gbps Bandwidth
  - 8 Optical Link (3Gbps each)
    - (upgrade: 6.5Gbps)
  - 5x Gigabit Ethernet
- 13x RocketIO to backplane (full mesh)
- 2 embedded PowerPC in each FPGA for slow control
- ATCA compliant



Compute Node, 2nd version







- IPMI remote control
  - Small piggy-back board
  - Functions: power on/off, power negotiation, health monitoring, board reset, bitstream selection

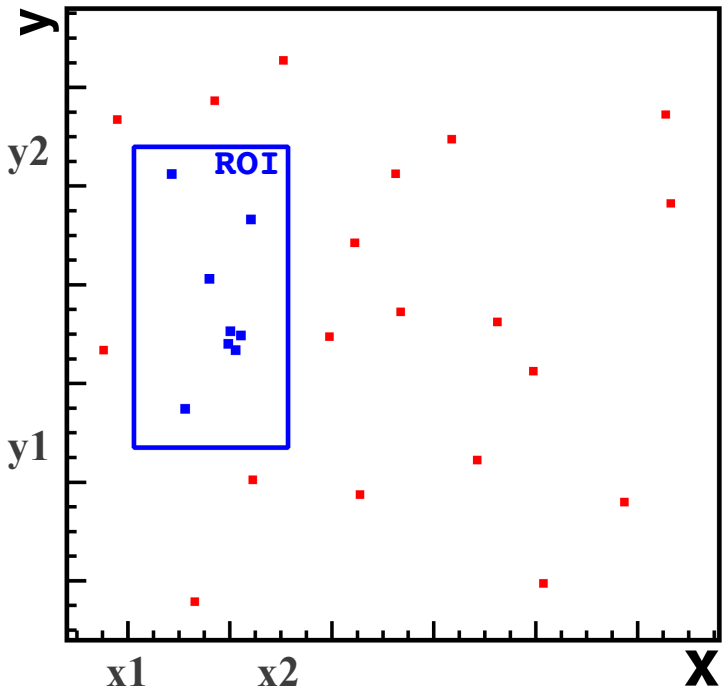
Talk by T. Geßler

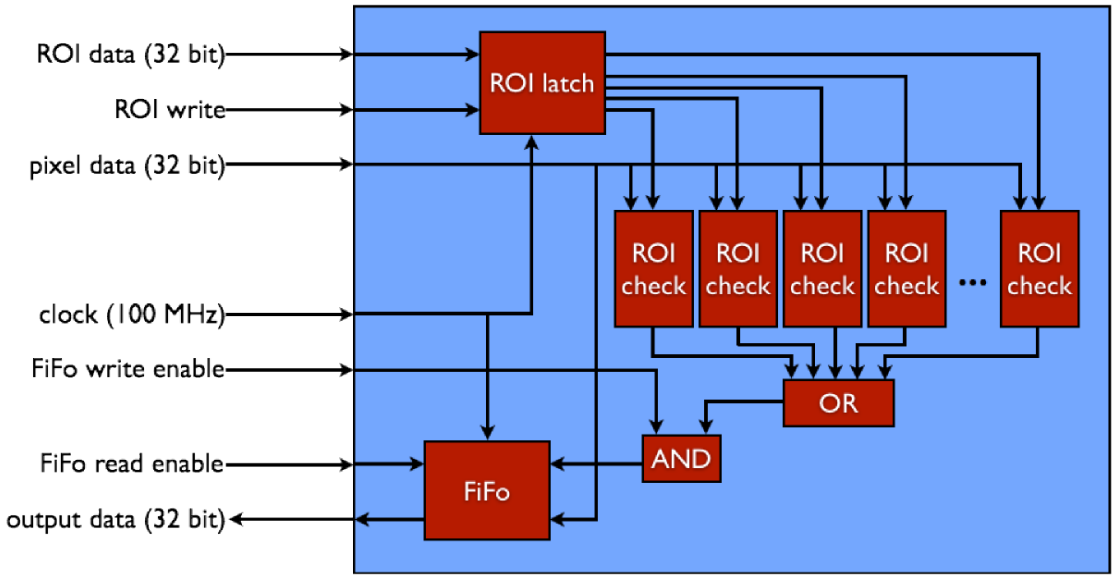
Expected data rates

- L=8×10^35 cm−2 s−1 => 30kHz event rate (maximum)
- Expected pixel occupancy: Mean 1% plus background (synchrotron radiation, beam-gas background, radiative QED processes, Touschek effect) and safety margin → 3%
  - **<u>≤180GBit/s</u>** for whole PXD detector (for 3%)
- DEPFET Pixel Detector: ~8M Pixels in (8+12)*2=40 Half Ladders
- For 40 optical links → 4.5GBit/s (550MB/s) per link
- **<u>Rate reduction</u>** by 1/5 (High Level Trigger, HLT) (save assumption from Belle experiences)
- **<u>Size reduction</u>** 1/10 per accepted event
  - Selection of pixels in ROI (Regions Of Interest)
  - ROIs from HLT and SVD tracklet

ROI Algorithm

- Region of Interest (ROI) selection
  - Aim: raw data reduction
  - ROIs calculated by SVD tracklet and HLT from CDC and SVD data
  - To check: x1<x<x2 and y1<y<y2
  - Simple algorithm, but parallelism (#ROIs) and throughput needed
  - A core with 31 ROIs is running @ 100MHz successful





Talk by D. Münchow

Possible extensions

- Improvements: low pT tracks
  - High dE/dx, track might not be seen by HLT (no CDC track)
  - Cluster finder
    - Calculate additional ROIs directly from pixel clusters
    - Tracking possible including PXD data?
  - SVD tracking
    - Calculate ROIs from SVD data

- Single photon / single pixel rejection
  - Isolated fired pixels
  - Saves bandwidth if removed before stored to RAM

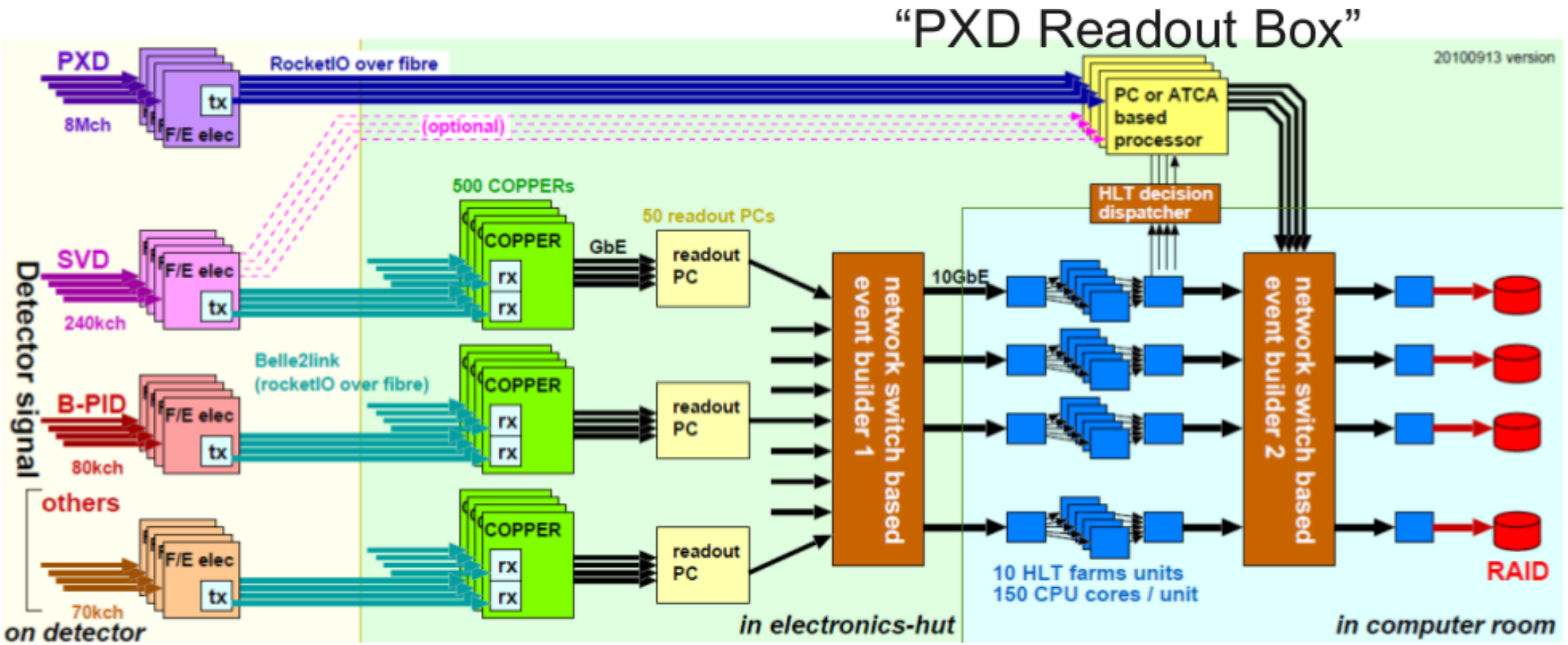
To be solved

- only 32 bit of 64 bit used (64 bit PLB but pixel 32 bit aligned)
  - two sub-cores? go back to 32 bit bus?
- cache: IP core updates memory, PPC does not notice due to data cache
  - invalidate/flush cache? not working.
  - not a problem for a real system
- FFF,FFF is always in (unused) ROI... not a problem at all
- write data to which buffer? offset? (kills ROI#32 at the moment)
- 8 bytes too much... because of 64 bit alignment and minimum copy for burst
- MPMC too slow?
- Software related:
  - TCP/IP server/client timeout(?) problem.

“expected” numbers

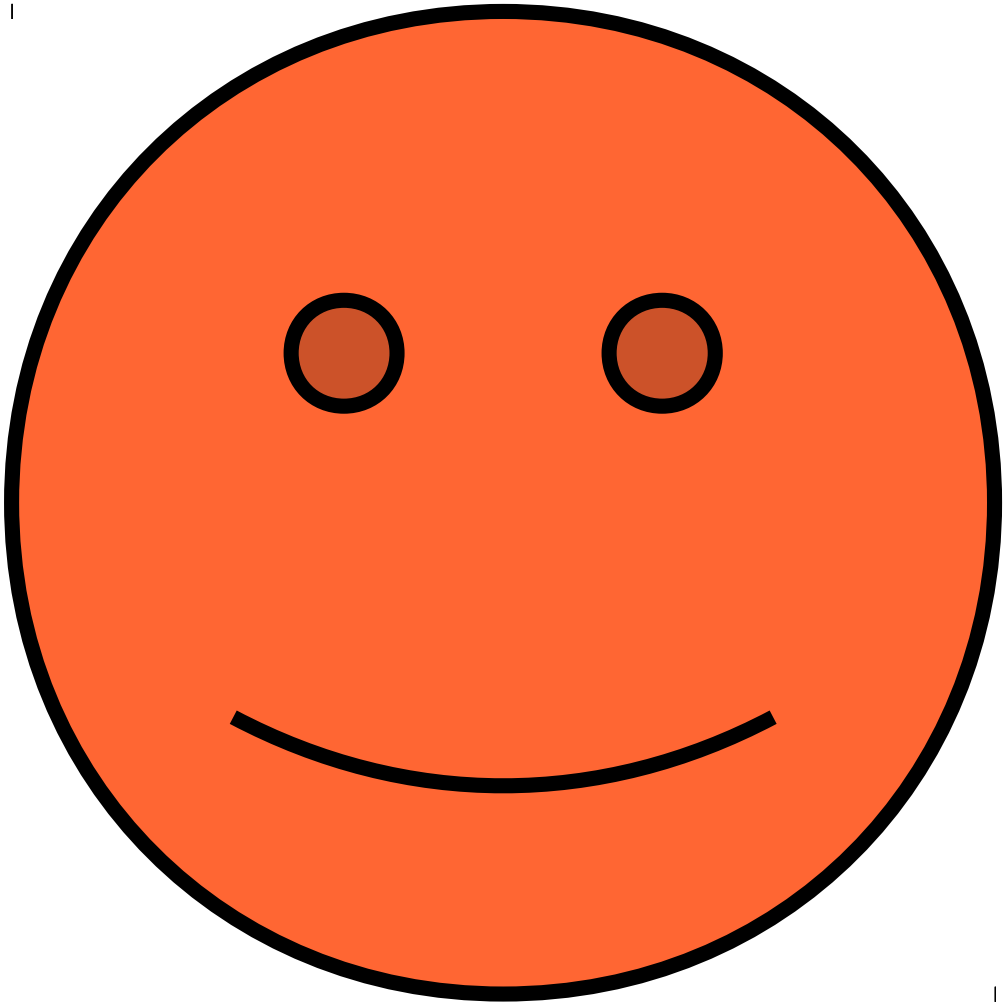
- max possible PLB: 64bit*100MHz/2(rw) = 400MB/s
- max possible MPMC 400MB read or 400MB write @ one PLB
-
- DDR2 @ 200MHz 64bit =>1600MB/s (read, write)
- PLB: 64bit 100MHz => 800MB/s (one direction)
  - 16ticks data 11ticks arb => ~475 max

Readout - Overview



- Foreseen for "PXD Readout Box"
  - 1 ATCA shelf (Advanced Telecommunication Computing Architecture)
  - 10 Compute Nodes with 4+1 FPGAs and 8 Optical Links

Summary



It Works!