FPGA Helix Tracking Algorithm for PANDA

Yutie Liang, Martin Galuska, Thomas Geßler, Wolfgang Kühn,
Jens Sören Lange, David Münchow, Björn Spruck, Milan Wagner

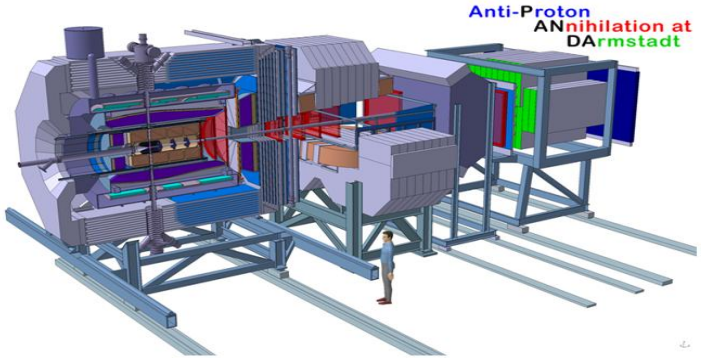
II. Physikalisches Institut, JUSTUS-LIEBIG-UNIVERSITÄT GIESSEN

International Conference on Science and Technology for FAIR in Europe 2014
Worms, 16.10.2014

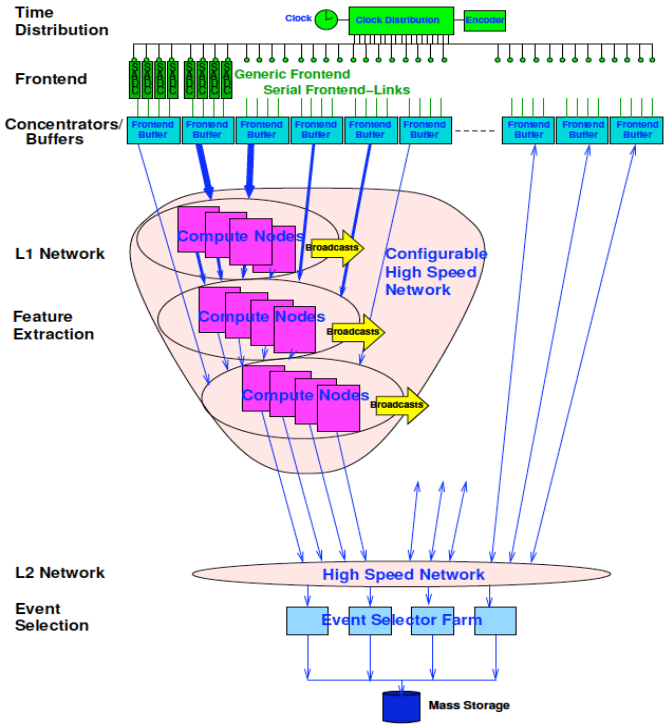
Outline

1. Introduction
   PANDA, DAQ, STT
2. Tracking algorithm
3. Performance study with C++
4. Implementation in VHDL
5. Summary and outlook

The Structure of PANDA TDAQ



- anti-proton of 15 GeV/c on proton fixed target
- Strong interaction studies with antiprotons
- $2*10^7$ annihilations /s, Trigger-less
- $2.5*10^4$ events/s to storage, online reduction factor of ~ 1000



➢**Time Distribution System** – provide clock for hit timestamps

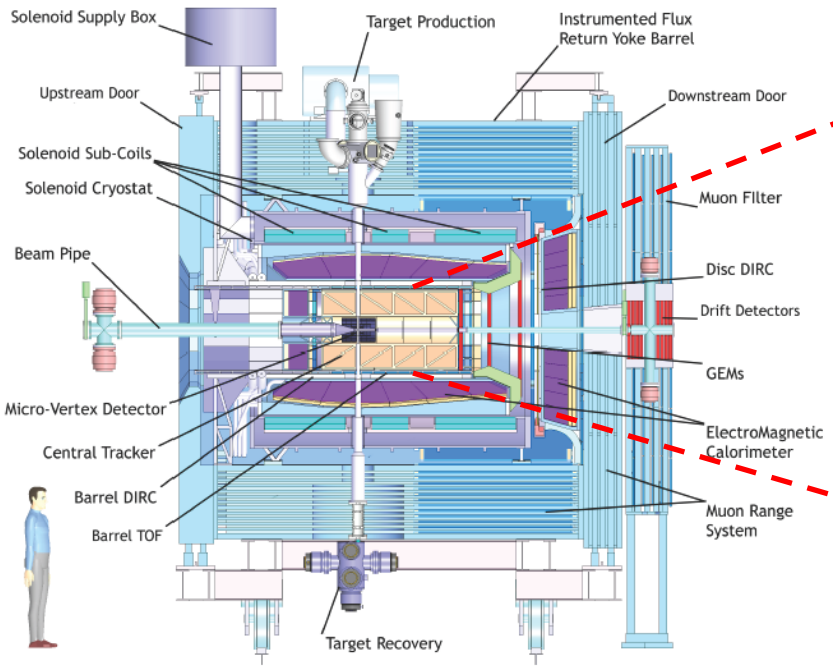
➢**Concentrators/Buffers** – buffering and on the fly data flow manipulation

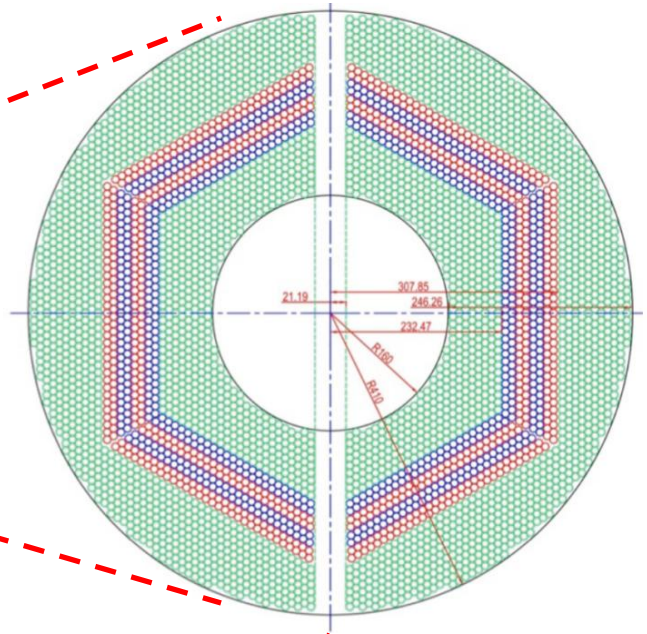
➢**L1 Compute Nodes** – mark hits which might belong to the same event (time slice)

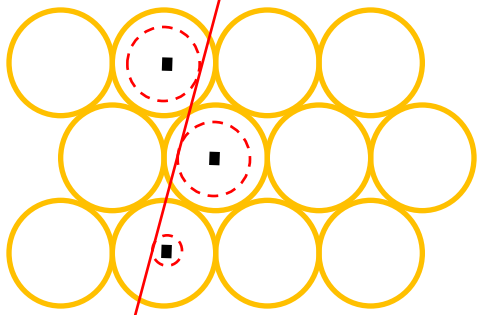
➢**L2 Feature Extraction Nodes** – combine detector information to extract physical signatures (momentum, …)

➢**L3 Event Selection Nodes** – event selection based on a complete reconstruction (PID, vertex, invariant mass, …)

Straw Tube Tracker (STT)







- ➢4636 Straw tubes
- ➢23-27 planar layers
- 15-19 axial layers (green) in beam direction
- 4 stereo double-layers for 3D reconstruction, with ±2.89 skew angle (blue/red)

From STT : Wire position + drift time

Event Structure -- pile-up

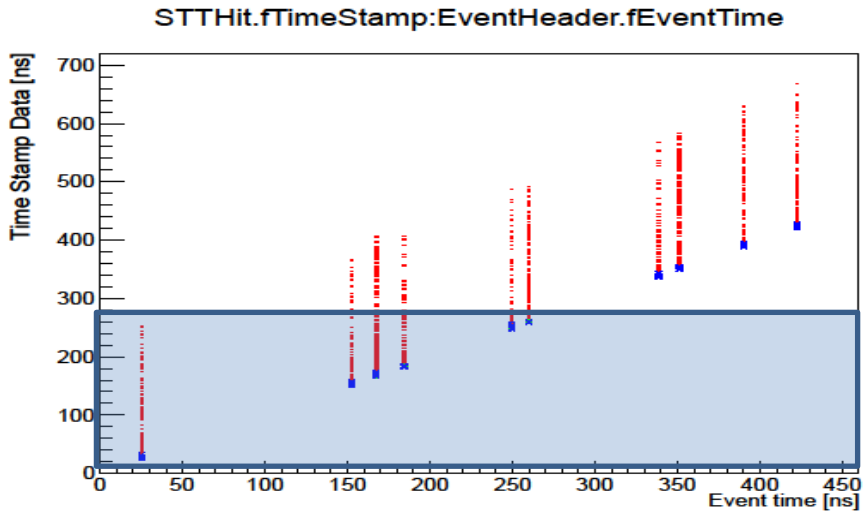
- 2000 ns revolution time
- 200 ns gap

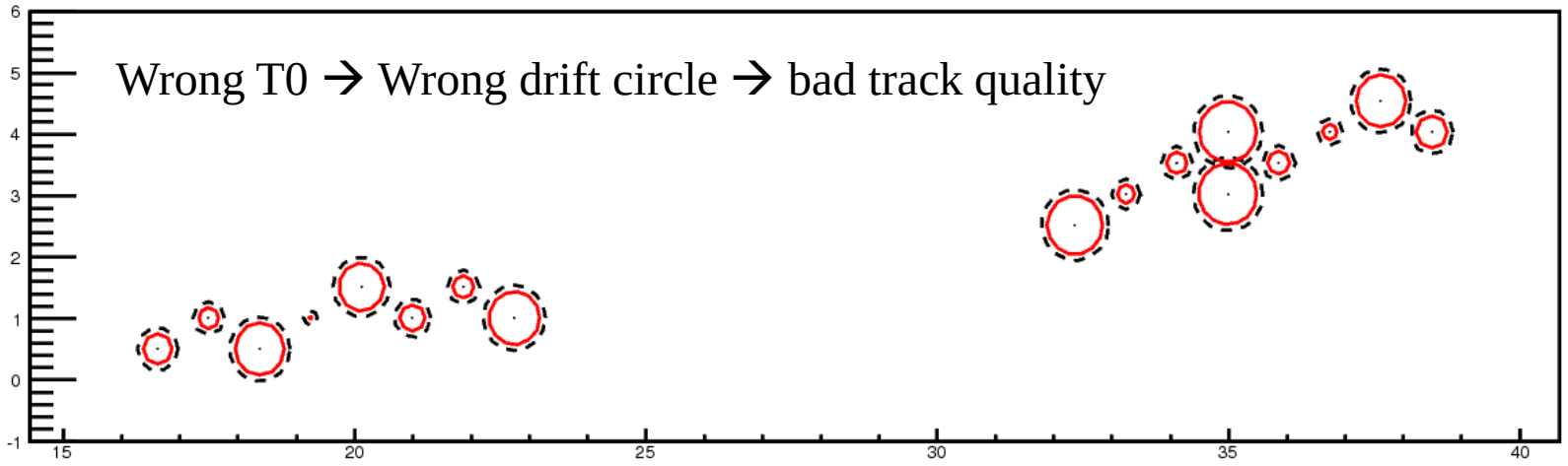
  In case of 20 MHz

➢ T_{Mean} between 2 events: 50 ns

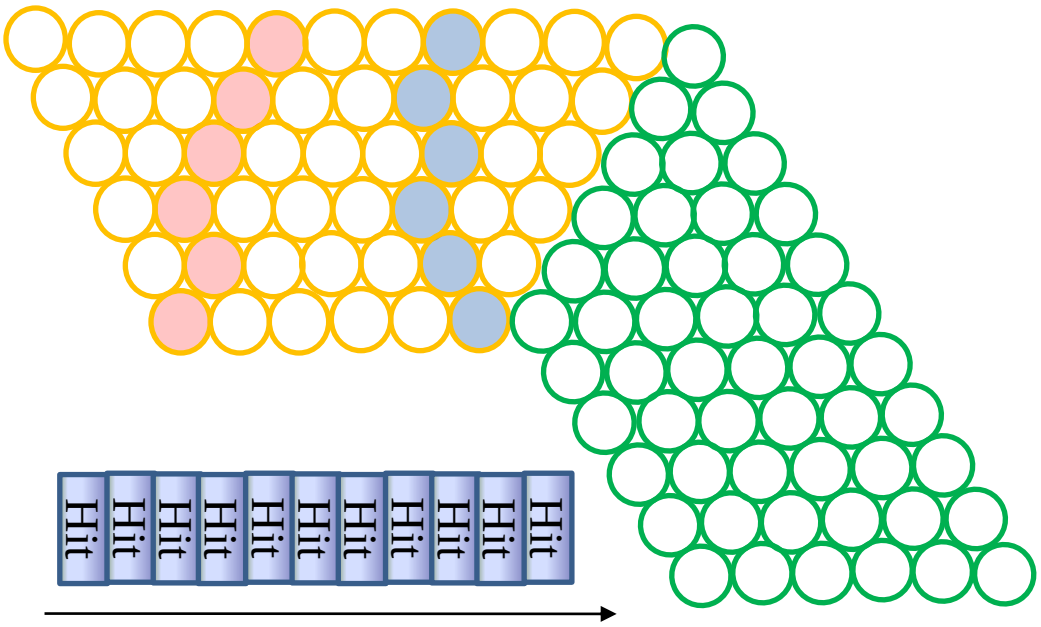
➢ Drift time: 0~200 ns

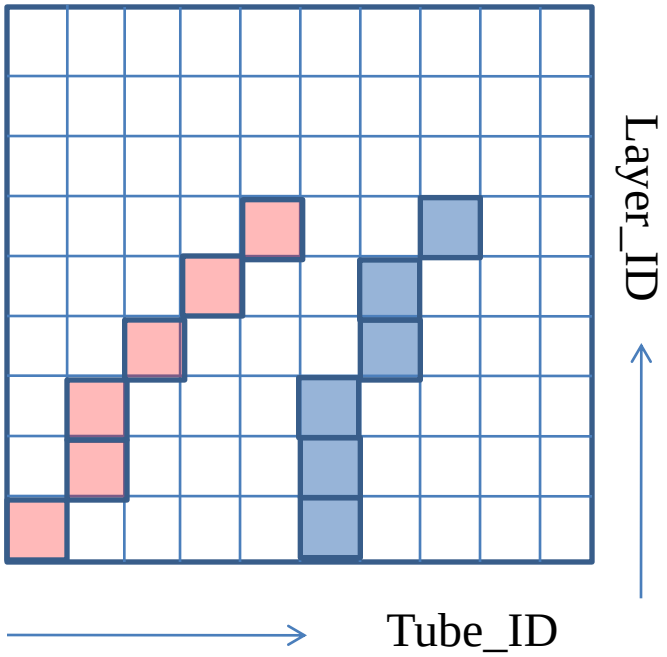
Event pile-up





Tracking Algorithm -- Road Finding





Hit: Seg_ID(3 bits) + LayerID(4 bits) + Tube_ID (6 bits) + Arrival time

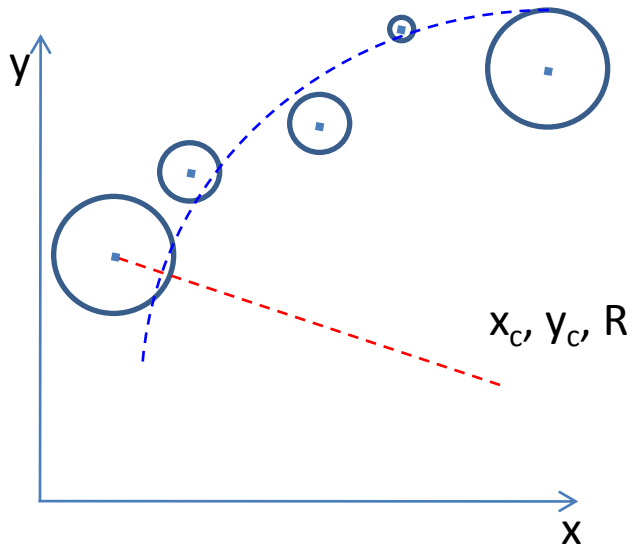
1: Start from inner layer

2: Attach neighbour hit to tracklet layer by layer

- ✓ Boundary between two segments.
- ✓ Number of neighbor:

  4 in axial layer; 6 in stereo layer

Tracking Algorithm -- helix parameters calculation



Known : x_i , y_i , d_i

Question: To determine a circle,

$$x^2 + y^2 + ax + by + c = 0$$

Method: Minimize the equation

$$E^2 = \sum(x_i^2 + y_i^2 + a\,x_i + b\,y_i + c)^2\,(1/d_i)^2$$

1) Circle para.

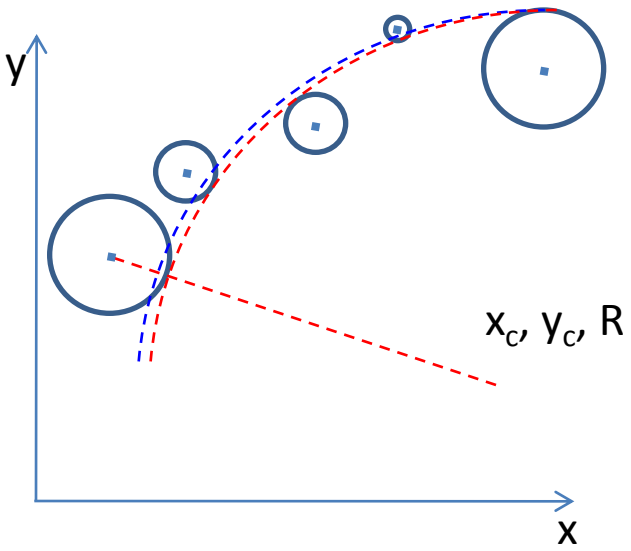
$$\begin{pmatrix} S_{xx} & S_{xy} & S_x \\ S_{xy} & S_{yy} & S_y \\ S_x & S_y & N \end{pmatrix} \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} -S_{xxx} - S_{xyy} \\ -S_{xxy} - S_{yyy} \\ -S_{xx} - S_{yy} \end{pmatrix}$$

$S_x = \sum x_i$ …
$S_{xx} = \sum x_i x_i$ …
$S_{xxx} = \sum x_i x_i x_i$ …

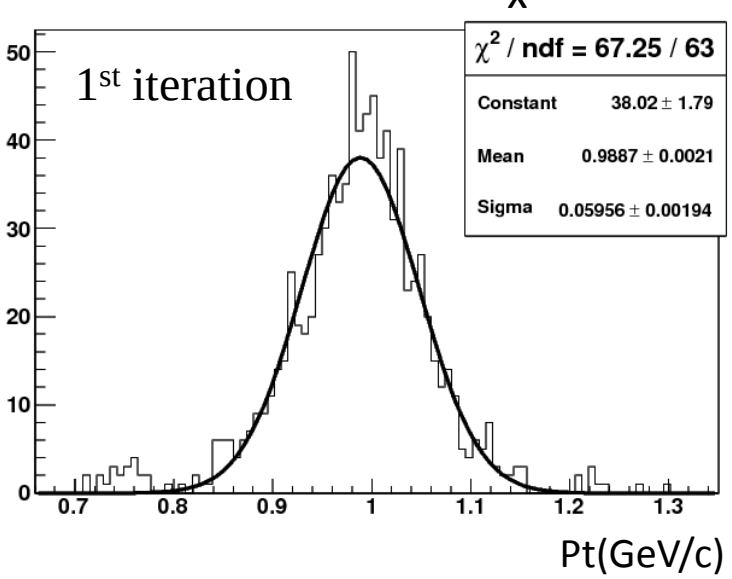
2) Track quality.

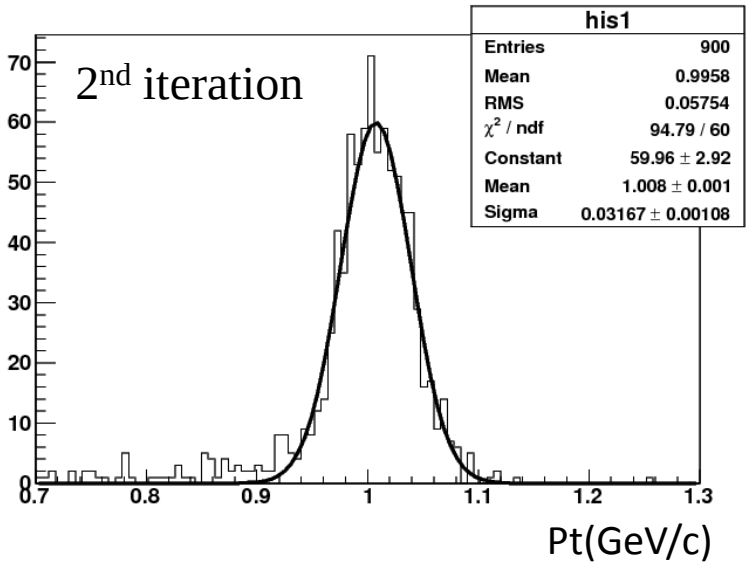
$$\chi^2 = 1/n \times \sum_i \frac{(x_i^2 + a x_i + y_i^2 + b y_i)/2r^2}{d_i^2}$$

To Improve the Momentum Resolution -- using a 2nd iteration



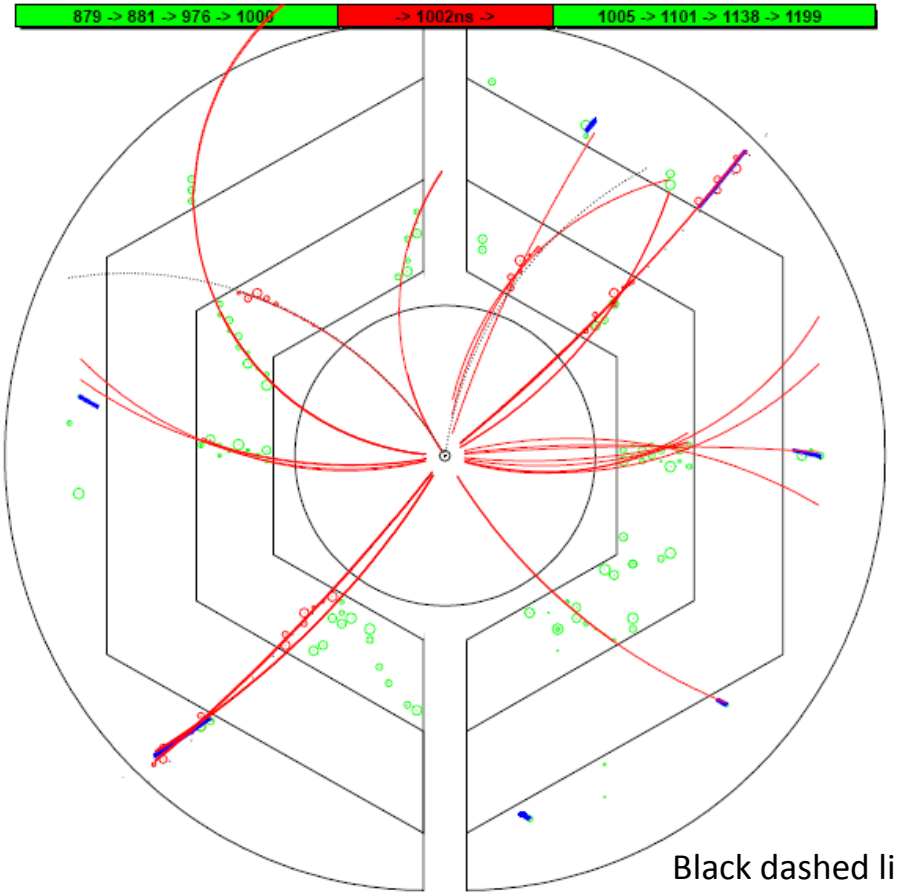
Pt(input)	1st iteration	2nd iteration
0.2GeV/c :	0.195 ±0.0068	0.195±0.0068
0.5GeV/c :	0.5 ±0.0212	0.5 ± 0.0164
1.0GeV/c :	0.99 ±0.0595	1.0 ± 0.0317
2.0GeV/c :	1.85 ±0.213	2.0 ± 0.073





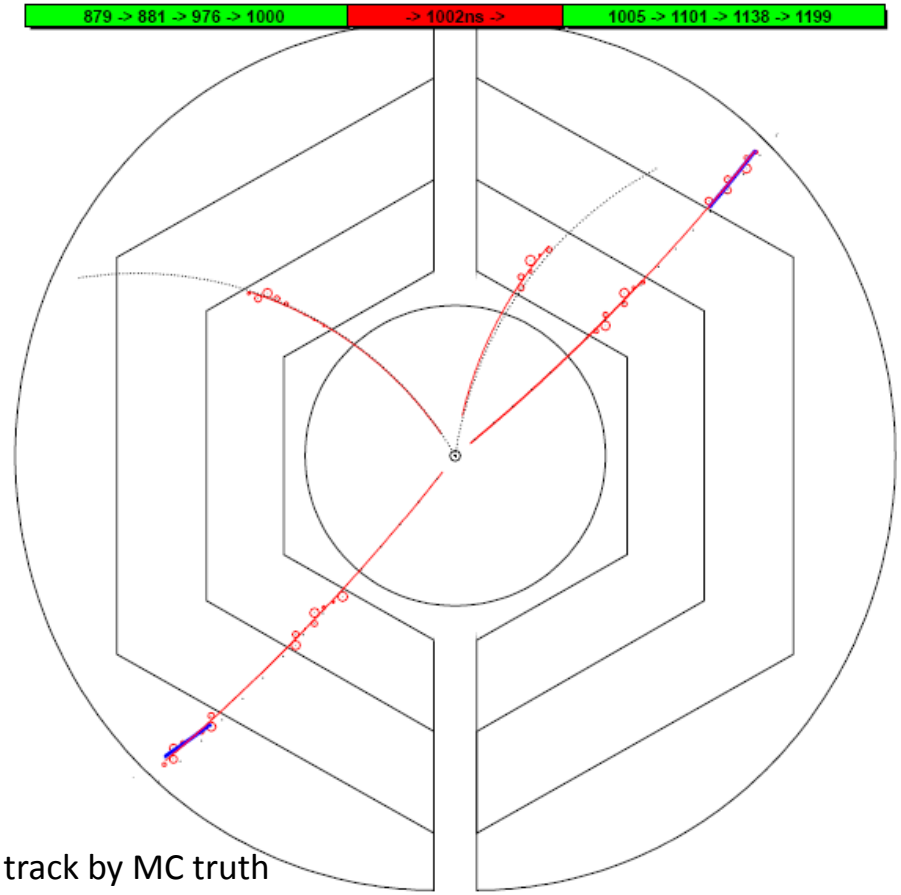
Performance Study -- realistic events, time-based

T_0 information: current event(Red) or other events(Green)



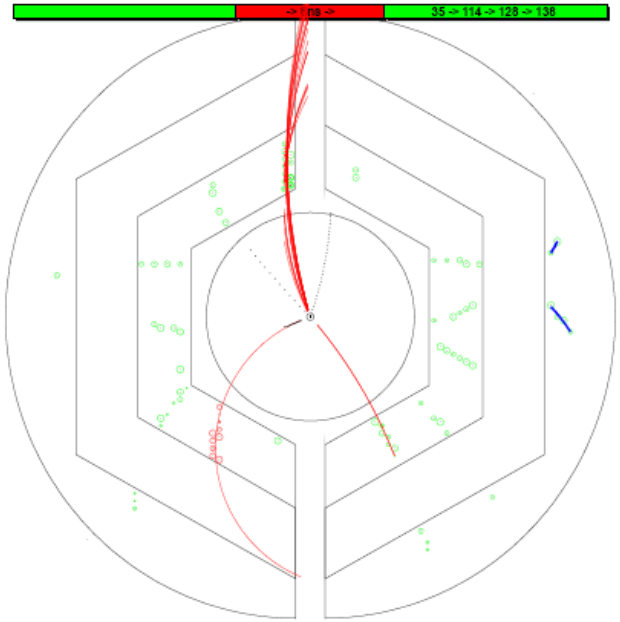
Event #4

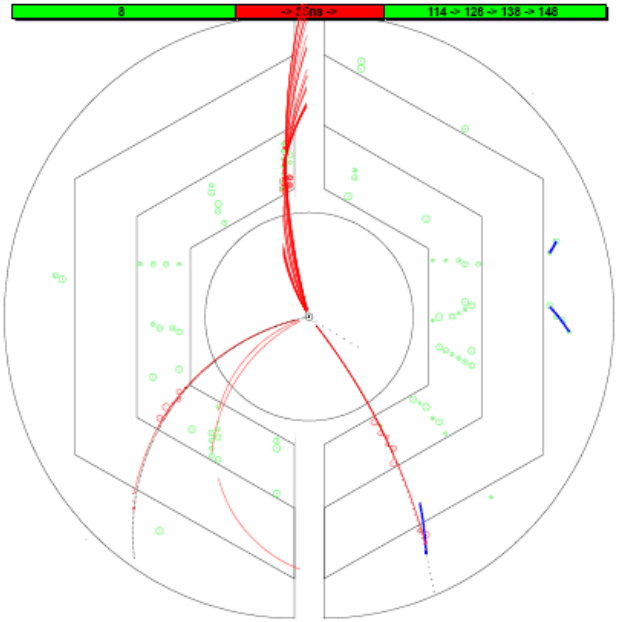
Black dashed line: track by MC truth
Red line: recon. track
Blue: recon. using outer layers
drift circles belong to current event (Red) or other events (Green)

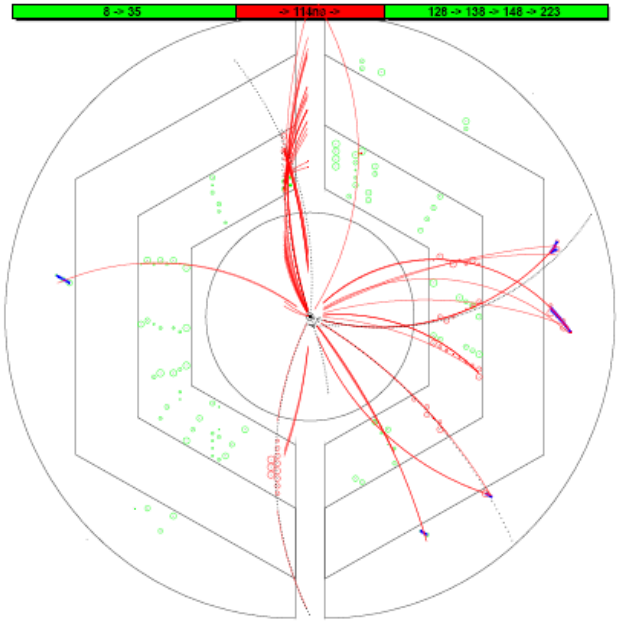


Event #4

Assign Track to Correct Event

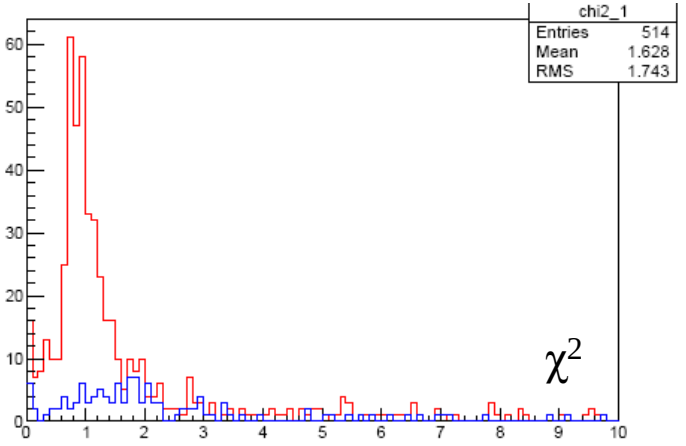






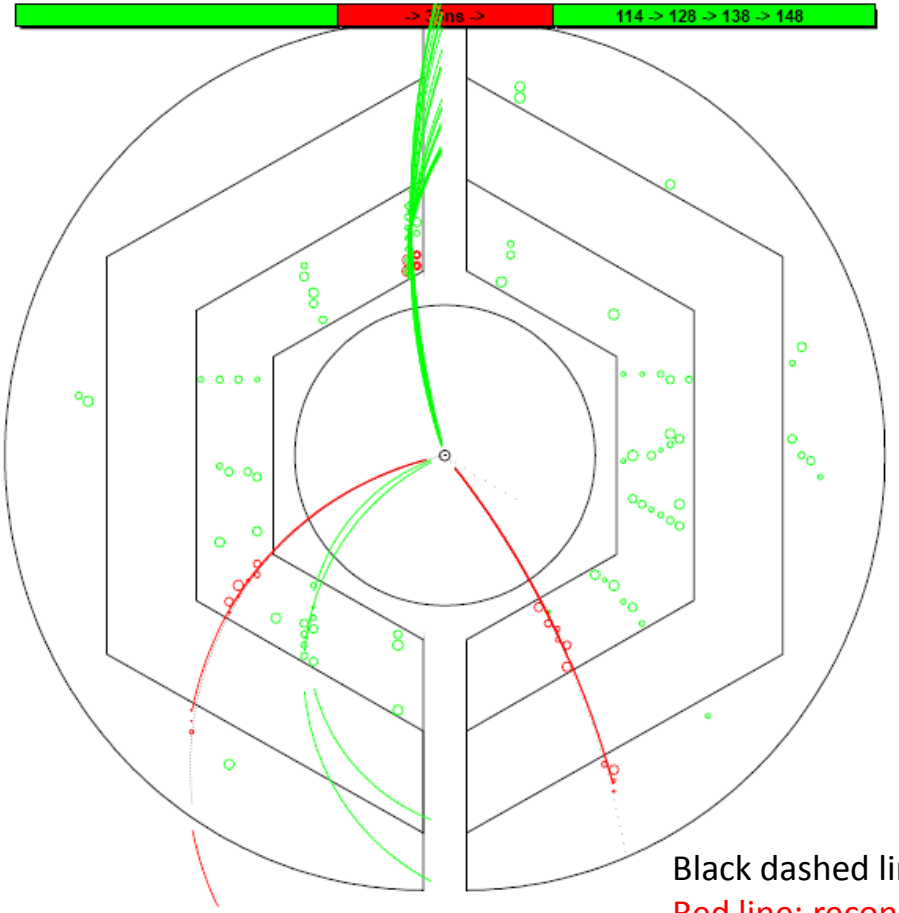
1: Number of hits in the track

2: χ^2 of the track

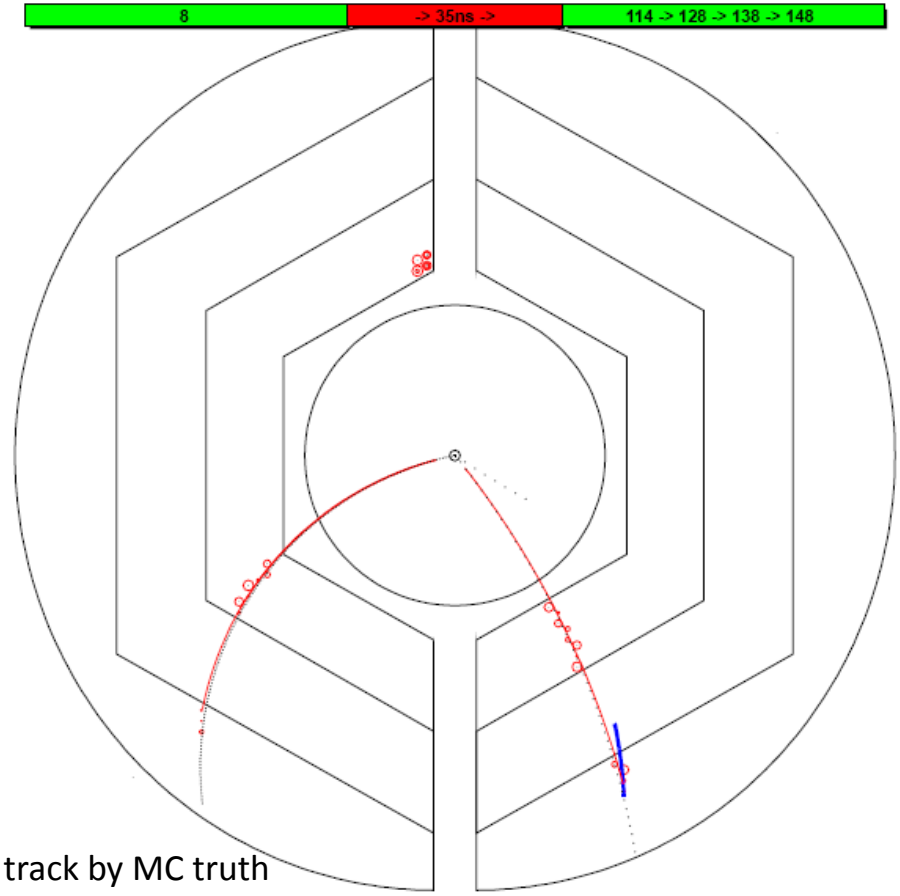


Assign Track to Correct Event

T_0 information: current event(Red) or other events(Green)



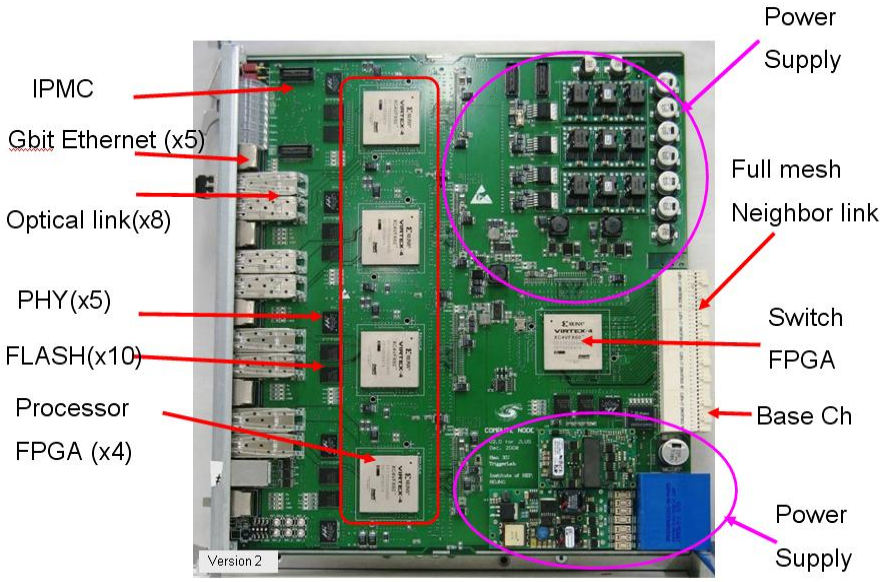
Event #1

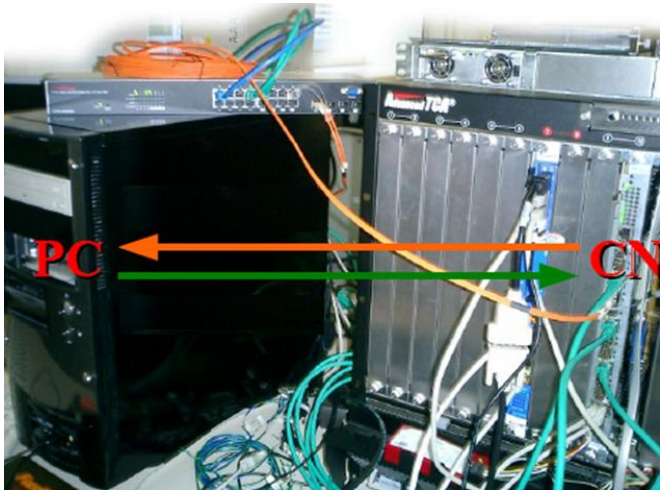


Event #1

Black dashed line: track by MC truth
Red line: recon. track
Blue: recon. using outer layers
drift circles belong to current event (Red) or other events (Green)

VHDL Implementation





VHDL: Very-high-speed integrated circuits Hardware Description Language.
It's a dataflow language, unlike procedural computing languages such as C++ which runs sequentially, one instruction at a time.

Performance at FPGA

Pz calculation not finished yet.

1: Device Utilization Summary:

Device Utilization Summary				
Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of Slice Flip Flops	17,187	50,560	33%	
DCM autocalibration logic	14	17,187	1%	
Number of 4 input LUTs	27,438	50,560	54%	
DCM autocalibration logic	8	27,438	1%	
Number of occupied Slices	18,159	25,280	71%	
Number of DSP48s	44	128	34%	

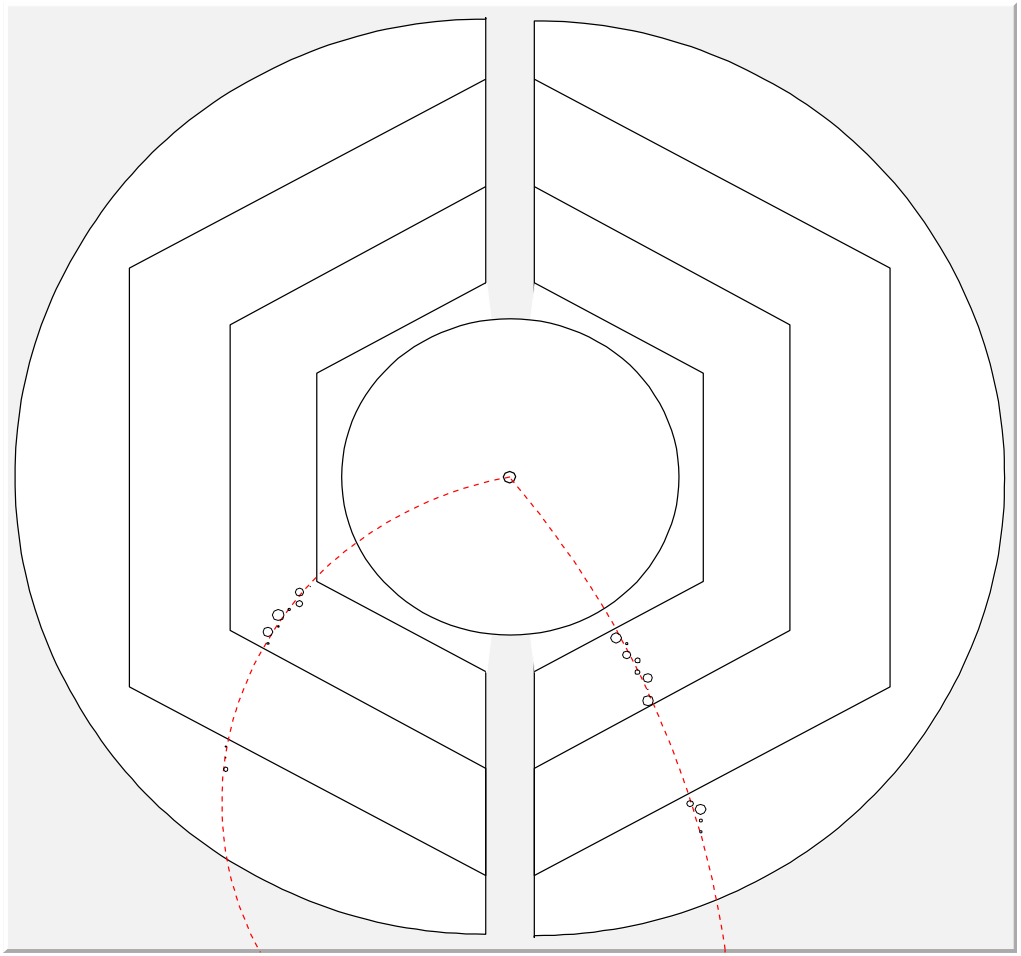
2: Time per event:

For one event with 100 hits (6 tracks):

~ 5 μs/event, 100 FPGA @ 20MHz interaction rate

Tracking at FPGA -- at low event rate

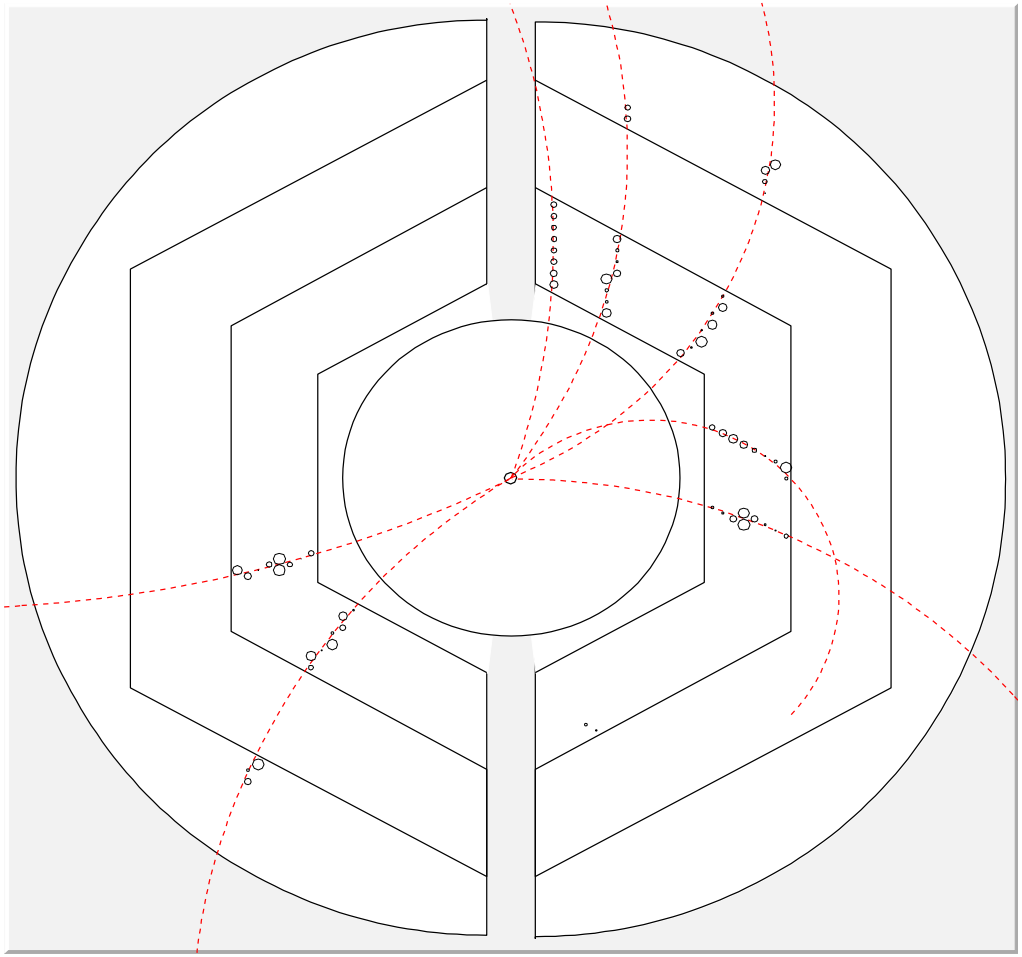
Data flow: Hit information at PC → FPGA, extract helix para. → PC



PC is used to draw hits and helix in the plot. The helix parameters come from FPGA.

Tracking at FPGA -- at low event rate

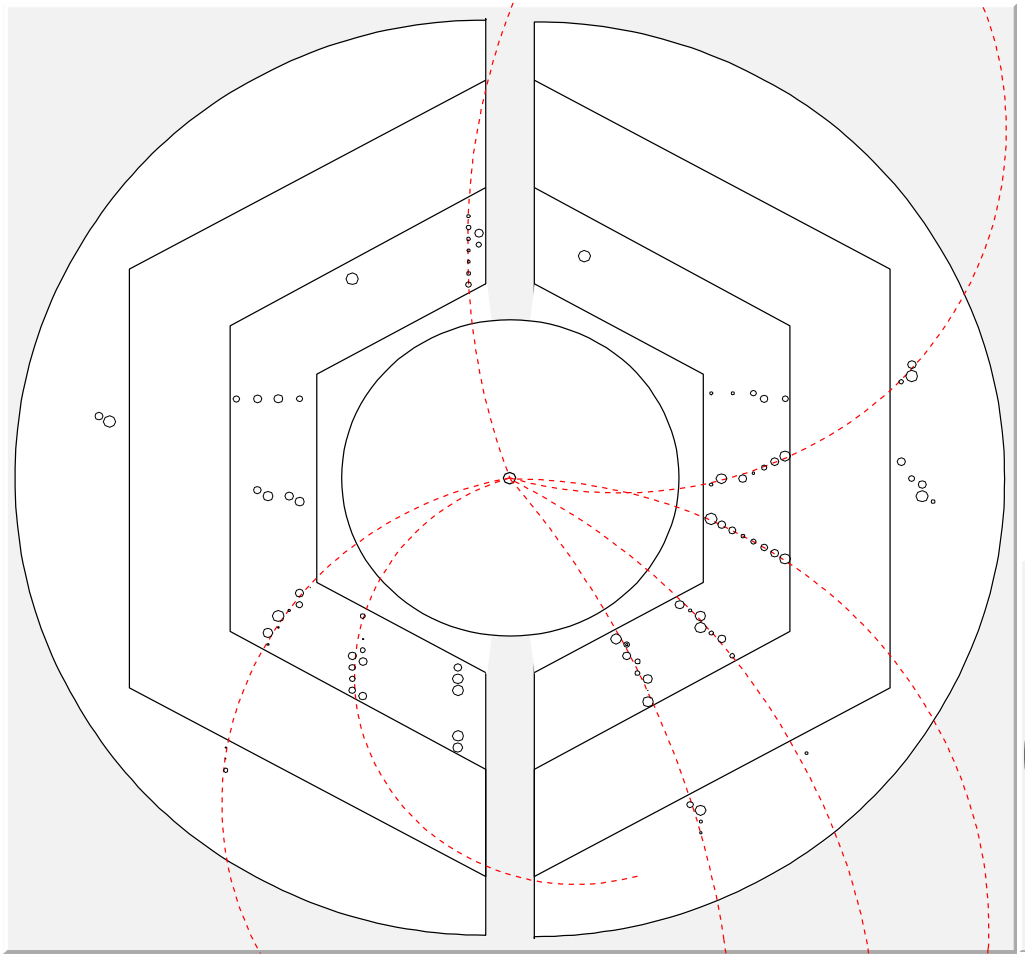
Data flow: Hit information at PC → FPGA, extract helix para. → PC

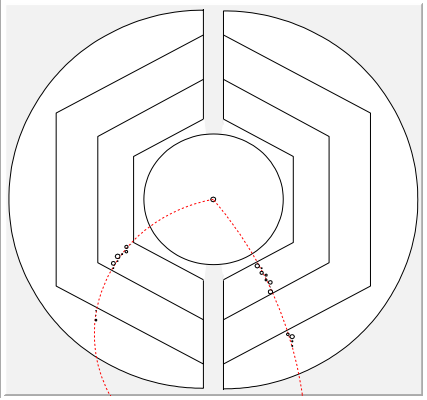


PC is used to draw hits and helix in the plot. The helix parameters come from FPGA.

Tracking at FPGA -- 20 MHz

Data flow: Hit information at PC → FPGA, extract helix para. → PC

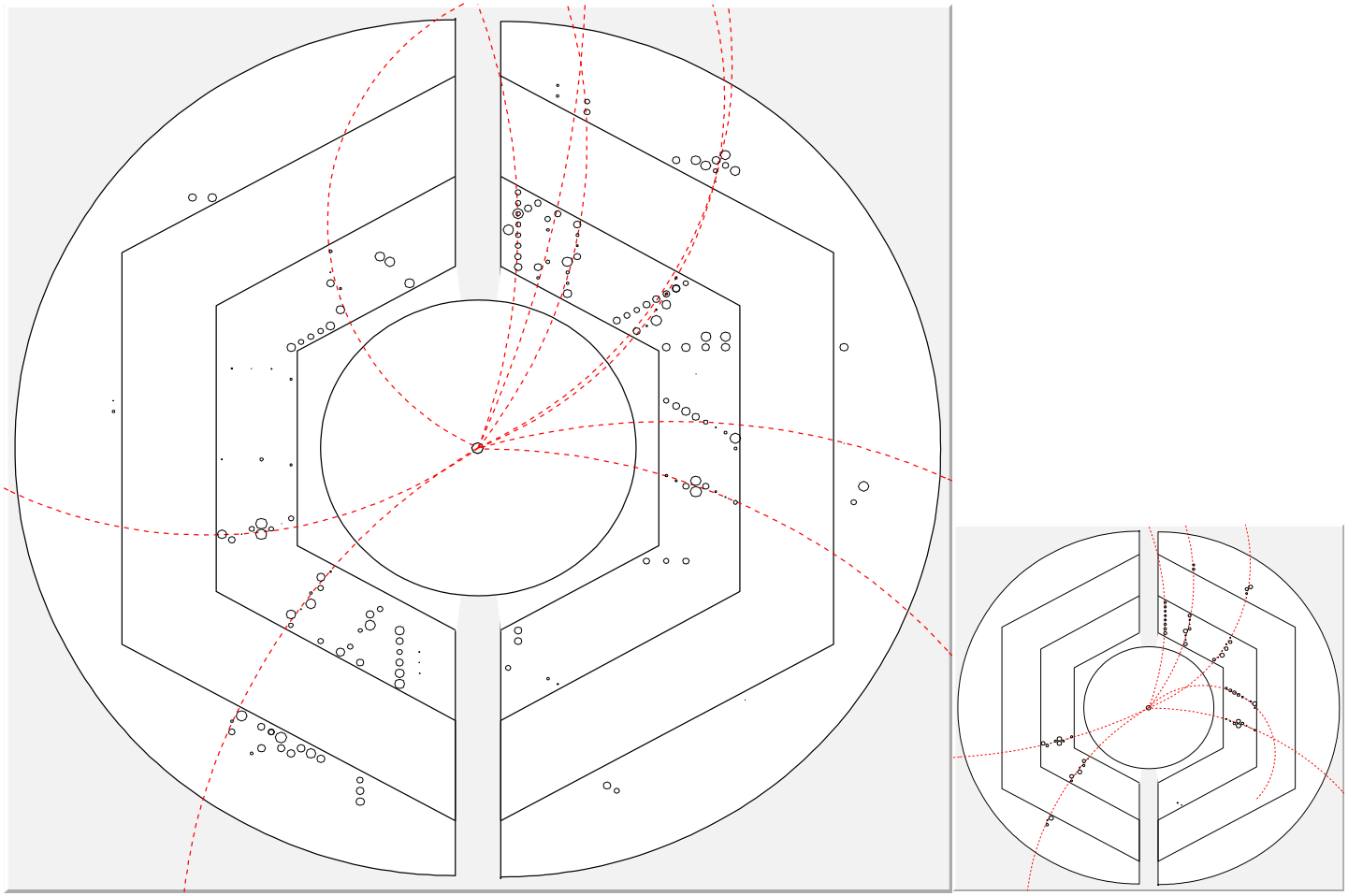




PC is used to draw hits and helix in the plot. The helix parameters come from FPGA.

Tracking at FPGA -- 20 MHz

Data flow: Hit information at PC → FPGA, extract helix para. → PC



PC is used to draw hits and helix in the plot. The helix parameters come from FPGA.

Summary and Outlook

- Performance of this tracking algorithm is studied using C++.
  - ✓ 2nd iteration to improve momentum resolution.
  - ✓ χ^2 of track might be used to determine T0.
- First VHDL code is working now.
  - ✓ ~ 5 μs/event, 100 FPGA @ 20MHz interaction rate

Next to do:

- Include MVD points.
- Test with PTDAQ.

Thank you

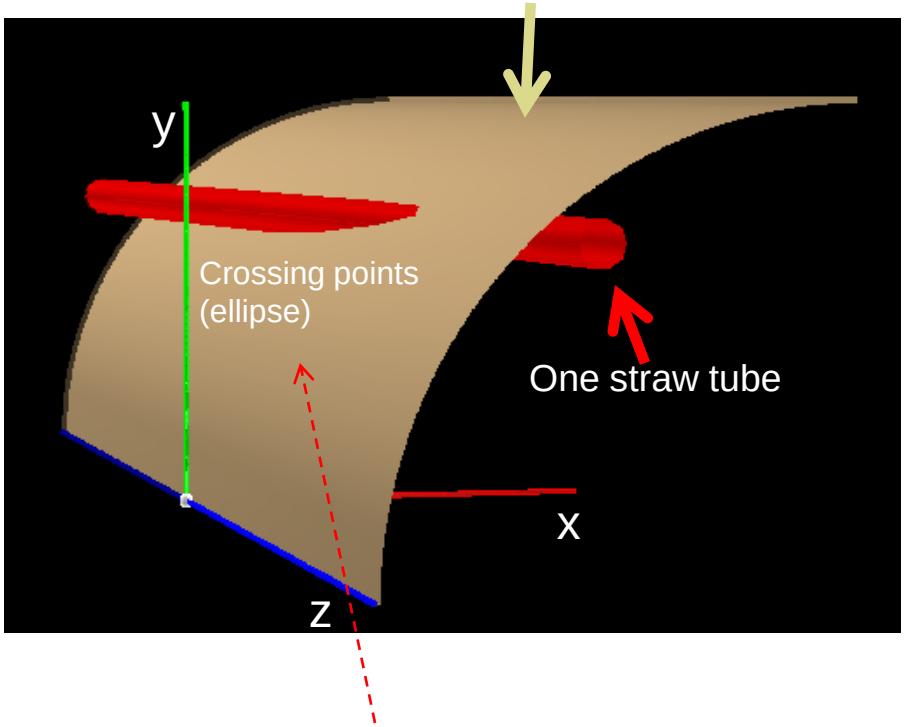
Back-up 1: Pz reconstruction

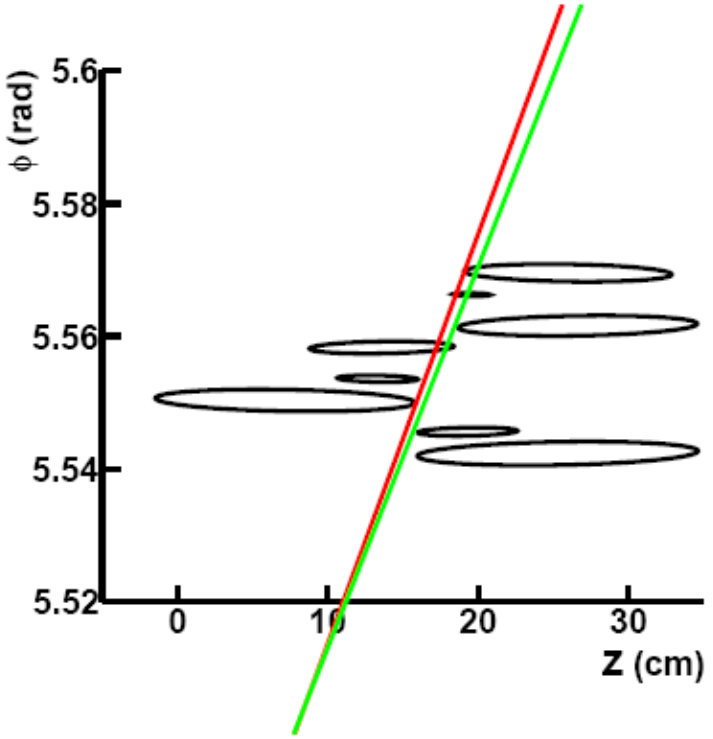
Pz reconstruction

1: The radius, the position of the helix center in the XY plane are determined.

2: $\Phi = kZ + \Phi_0$

One cylinder the helix lies in.
Based on Pt determined before

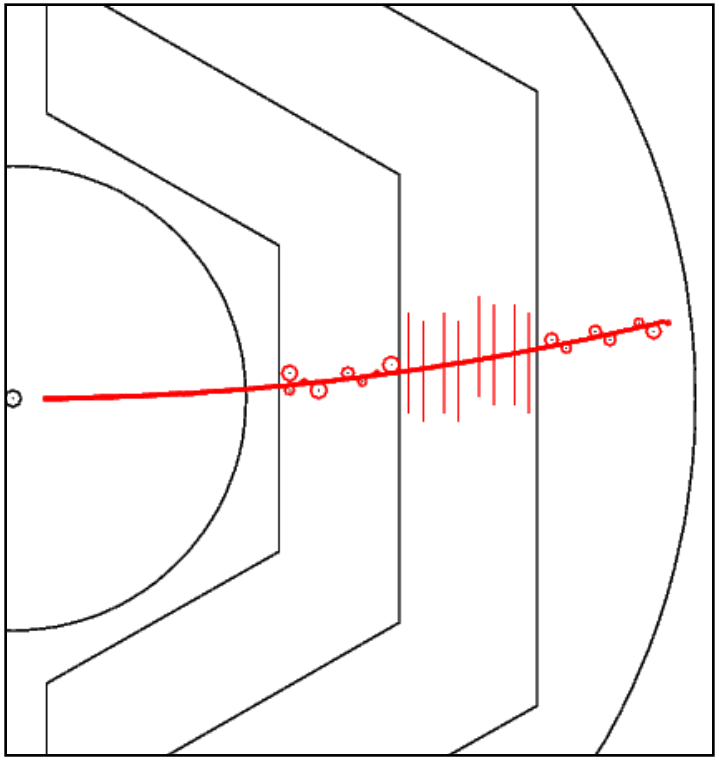


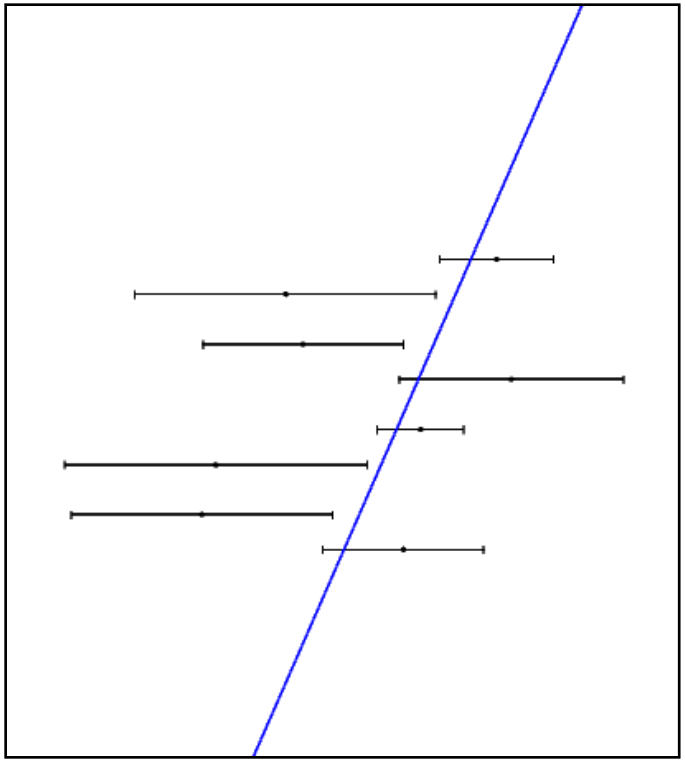


Track need to be tangent to the crossing ellipse.

Gianluigi Boca and Panda Collaboration, Panda STT TDR, 2012

Pz reconstruction





Known : z_i , Φ_i , d_i

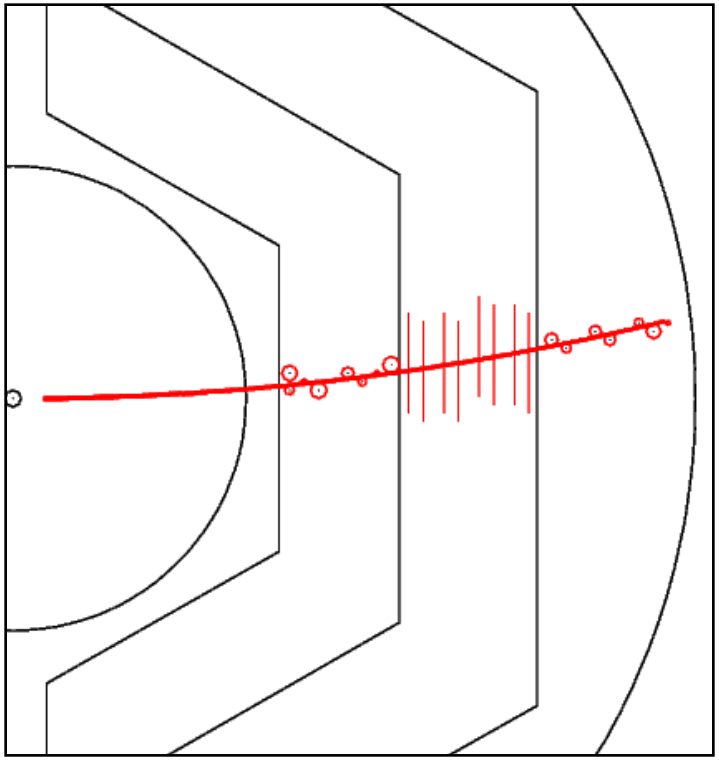
Question: To determine a line,

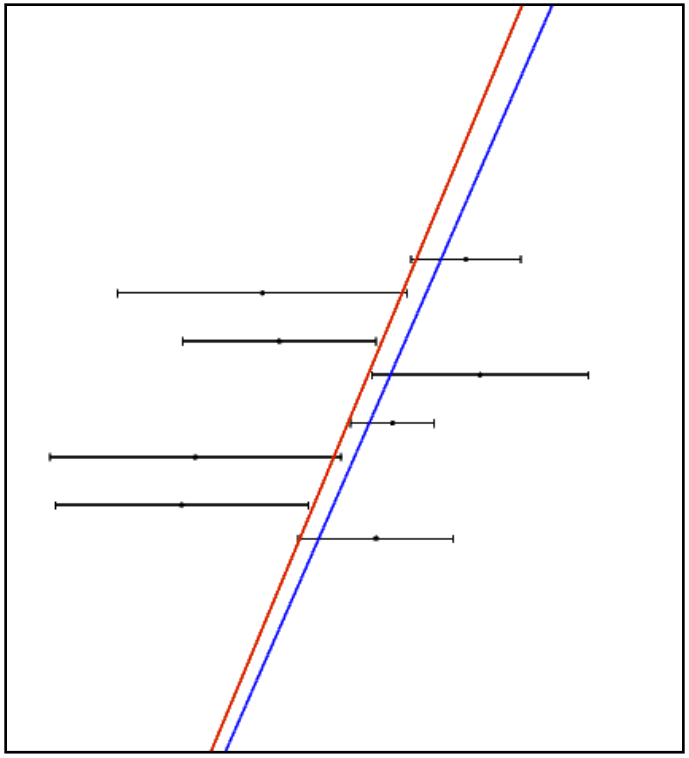
$\Phi + kz + \Phi_0 = 0$

Method: Minimize $E^2 = \sum(\Phi_i + kz_i + \Phi_0)^2 (1/d_i)^2$

$$\begin{pmatrix} S_{zz} & S_z \\ S_z & 1 \end{pmatrix} \begin{pmatrix} k \\ \phi_0 \end{pmatrix} = \begin{pmatrix} -S_{\phi z} \\ -S_\phi \end{pmatrix}$$

Pz reconstruction





Pt = 0.5 GeV/c

Pz(input)
0.25 GeV/c : 3.7 %
0.50 GeV/c : 4.0 %
1.00 GeV/c : 4.3 %

Pt = 1.0 GeV/c

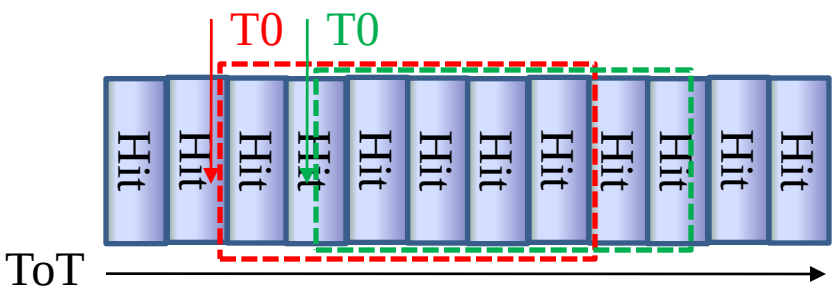
Pz(input)
0.5 GeV/c : 3.1 %
1.0 GeV/c : 3.8 %
2.0 GeV/c : 4.2 %

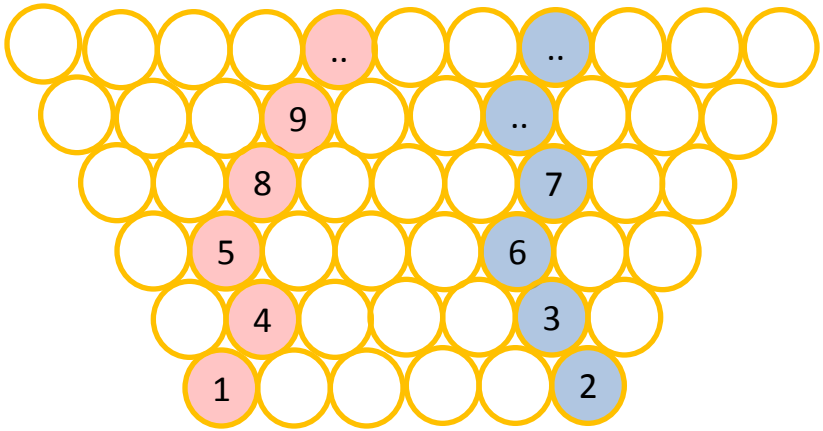
Pt = 2.0 GeV/c

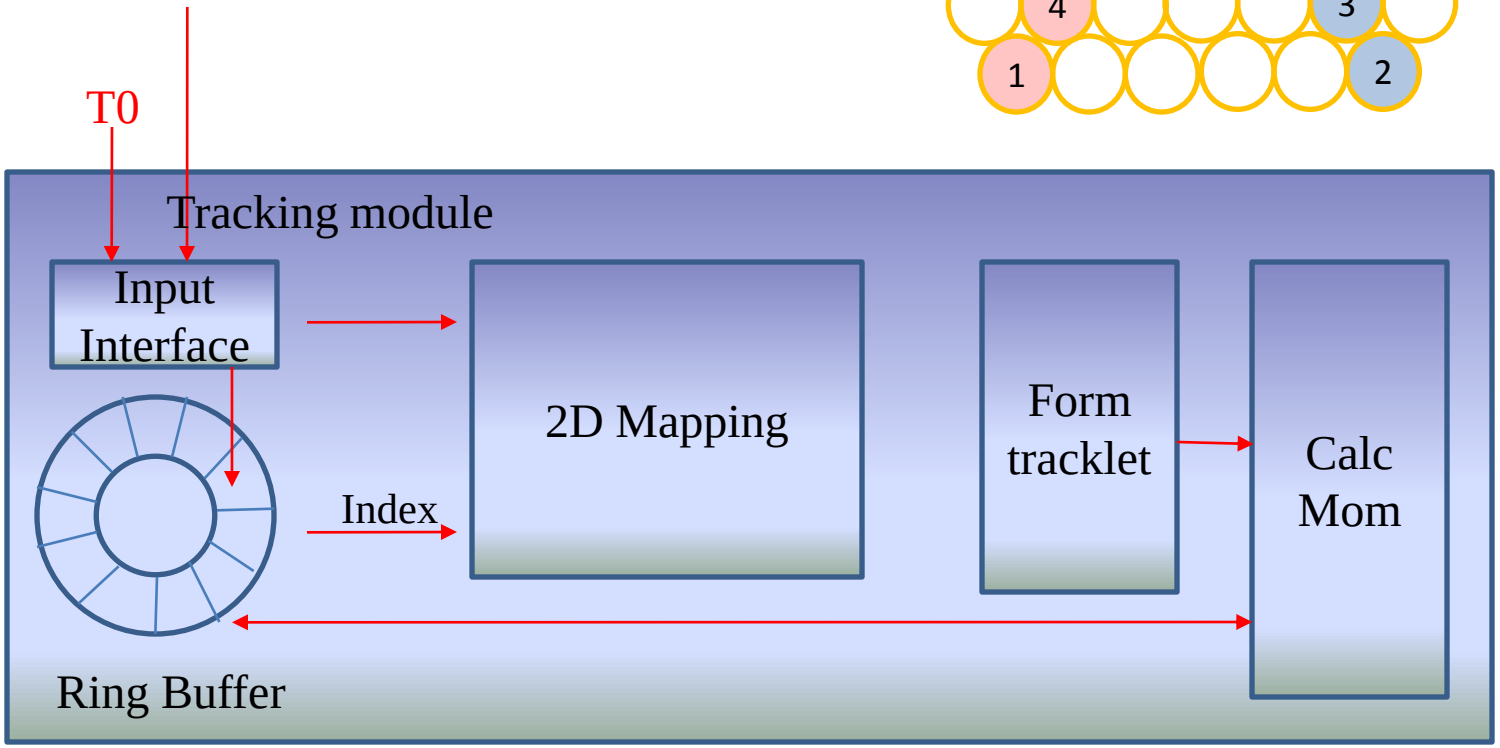
Pz(input)
1.0 GeV/c : 3.7 %
2.0 GeV/c : 4.8 %
4.0 GeV/c : 5.2 %

Device Utilization Summary				
Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of Slice Flip Flops	17,354	50,560	34%	
DCM autocalibration logic	14	17,354	1%	
Number of 4 input LUTs	21,326	50,560	42%	
DCM autocalibration logic	8	21,326	1%	
Number of occupied Slices	14,708	25,280	58%	
Number of Slices containing only related logic	14,708	14,708	100%	
Number of Slices containing unrelated logic	0	14,708	0%	
Total Number of 4 input LUTs	21,877	50,560	43%	
Number used as logic	19,854			
Number used as a route-thru	551			
Number used as 16x1 RAMs	8			
Number used as Shift registers	1,464			
Number of bonded IOBs	36	576	6%	
IOB Flip Flops	3			
IOB Dual-Data Rate Flops	1			
Number of BUFG/BUFGCTRLs	6	32	18%	
Number used as BUFGs	5			
Number used as BUFGCTRLs	1			
Number of FIFO16/RAMB16s	112	232	48%	
Number used as RAMB16s	112			
Number of DSP48s	124	128	96%	
Number of DCM_ADVs	2	12	16%	
Average Fanout of Non-Clock Nets	3.09			

VHDL implementation

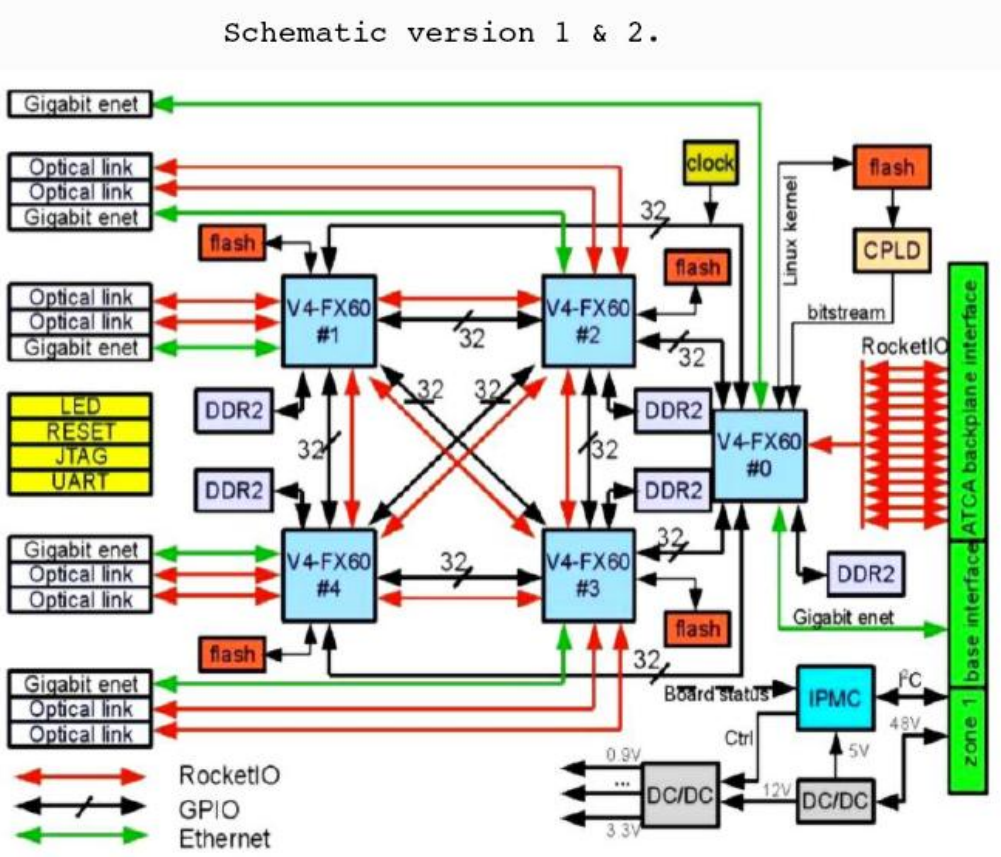






The Compute Node (CN)

- High Performance Computing
  - 5 Virtex-4 FX60 FPGA
    - (upgrade: Virtex 5)
  - 5*2 GB DDR2 RAM
    - (upgrade: 4 GB)
  - interconnected by RocketIO
- ~32 Gbps Bandwidth
  - 8 Optical Link (3.125 Gbps each)
    - (upgrade: 6.5 Gbps)
  - 5x Gbit Ethernet
- 13x RocketIO to backplane (full mesh)
- 2 embedded PowerPC in each FPGA for slow control
- ATCA compliant (Advanced Telecommunications Computing Architecture)

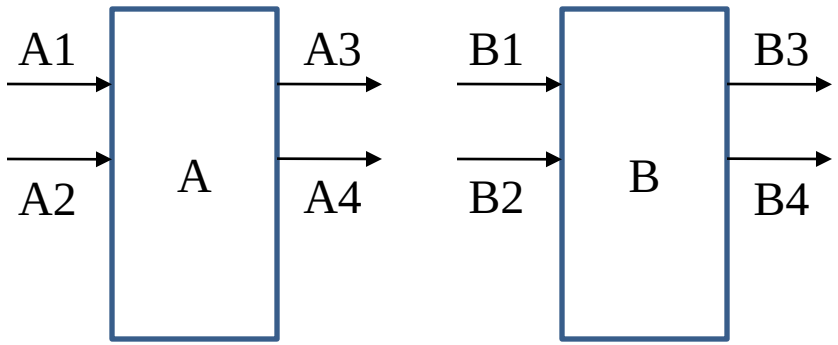


Hardware Description Language

VHDL, Verilog, SystemC

VHDL: Very-high-speed integrated circuits Hardware Description Language. It's a dataflow language, unlike procedural computing languages such as C++ which runs sequentially, one instruction at a time.

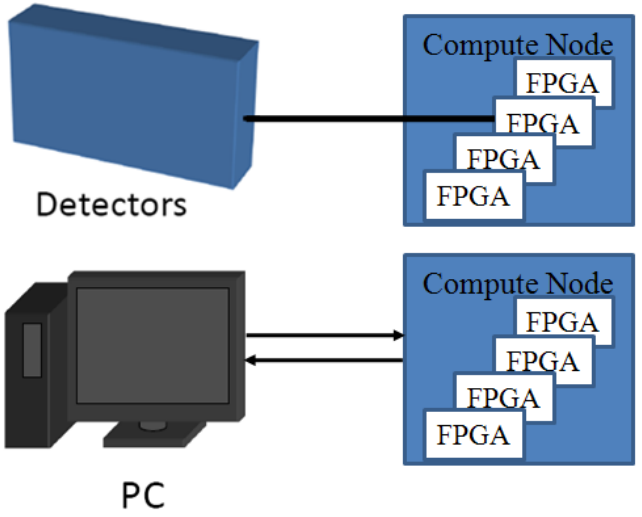
```
Signal A3_B1 …;
Signal A4_B2 …;

A3 <= A3_B1;
B1 <= A3_B1;

A4 <= A4_B2;
B2 <= A4_B2;
```

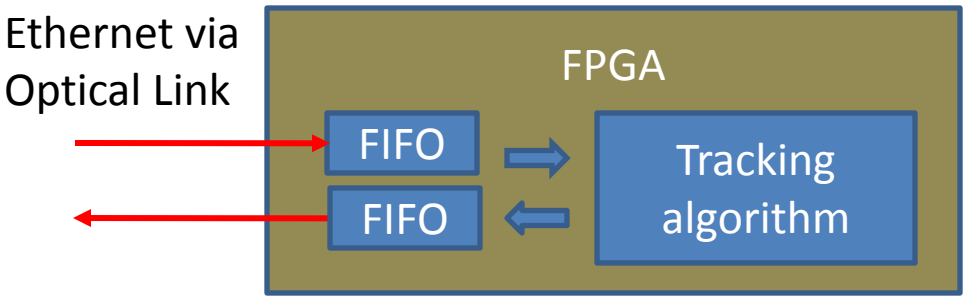


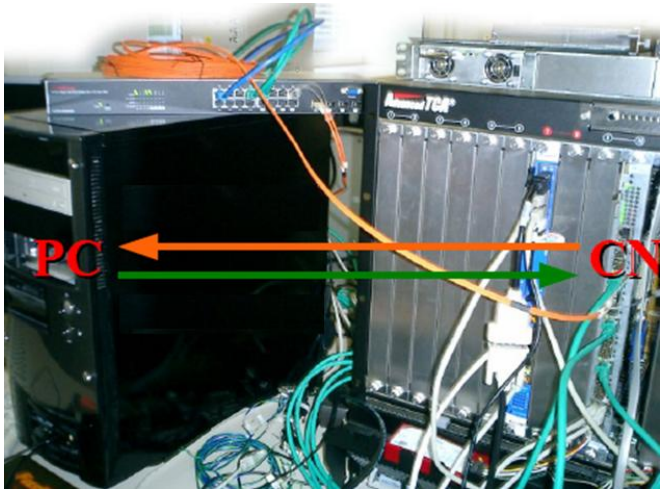
Setup and Test

PC as data source and receiver.

- Ethernet.
- Optical link (UDP by Grzegorz Korcyl) (not integrated yet)

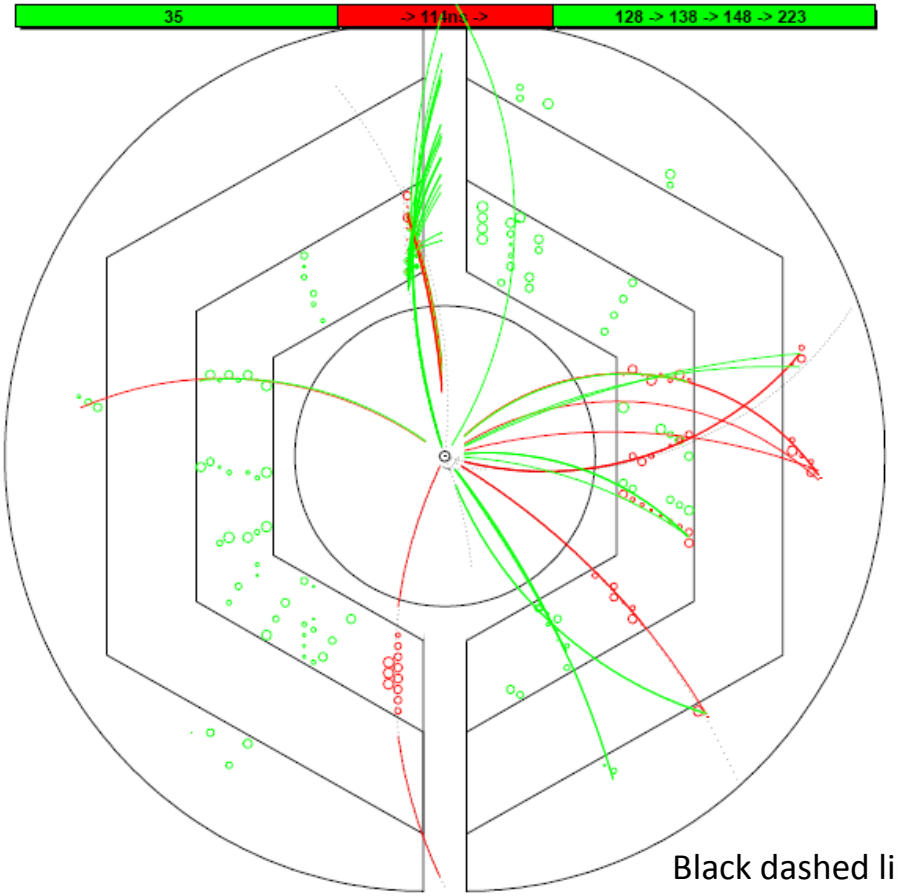






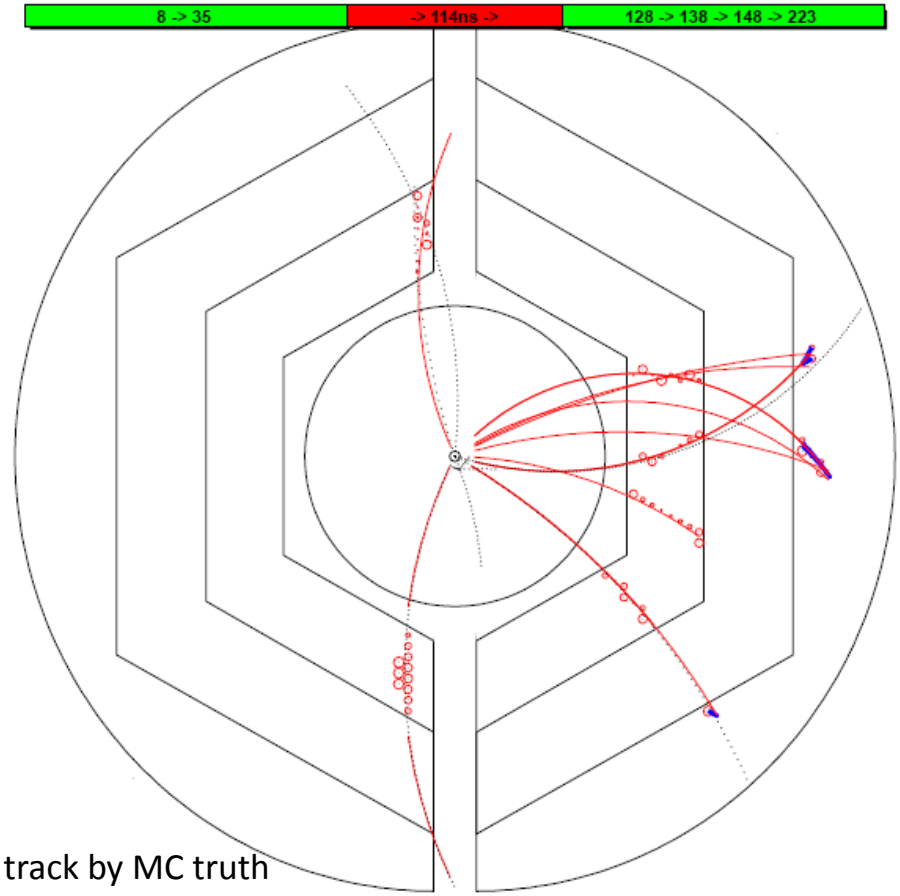
Assign a track to the correct event

T_0 information: current event(Red) or other events(Green)



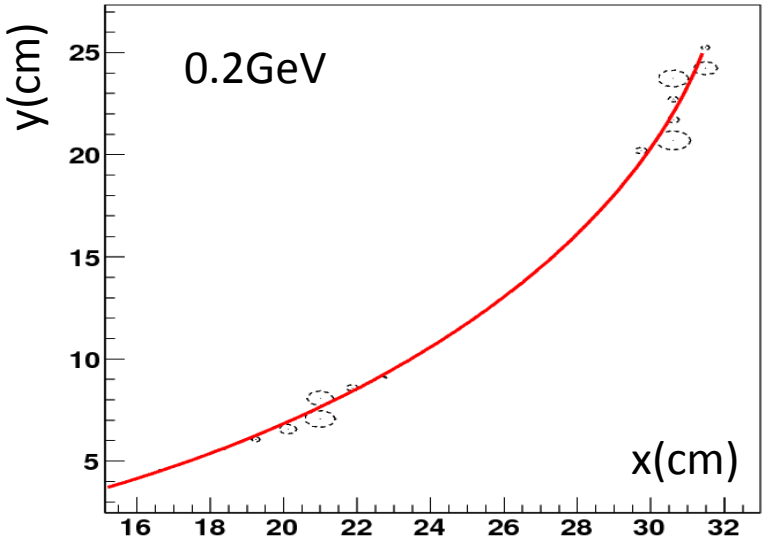
Event #2

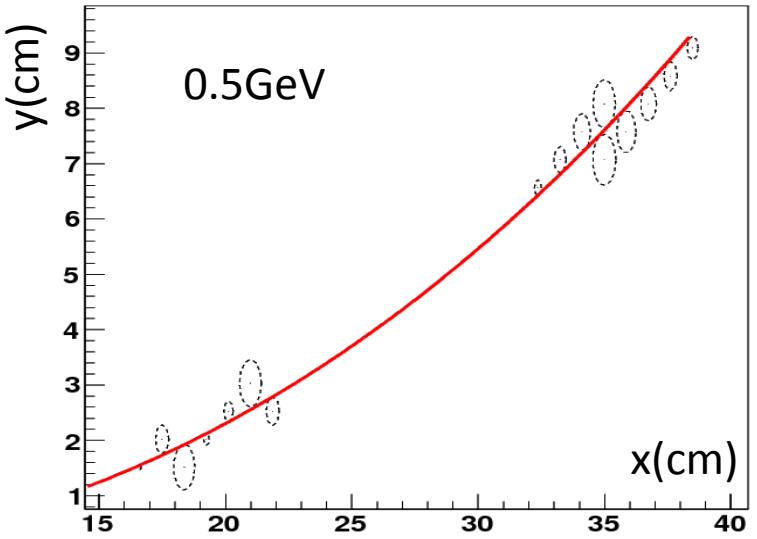
Black dashed line: track by MC truth
Red line: recon. track
Blue: recon. using outer layers
drift circles belong to current event (Red) or other events (Green)

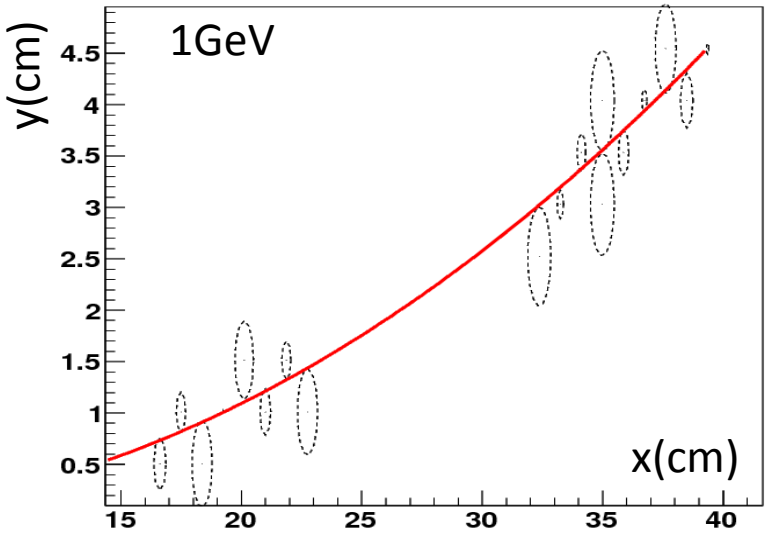


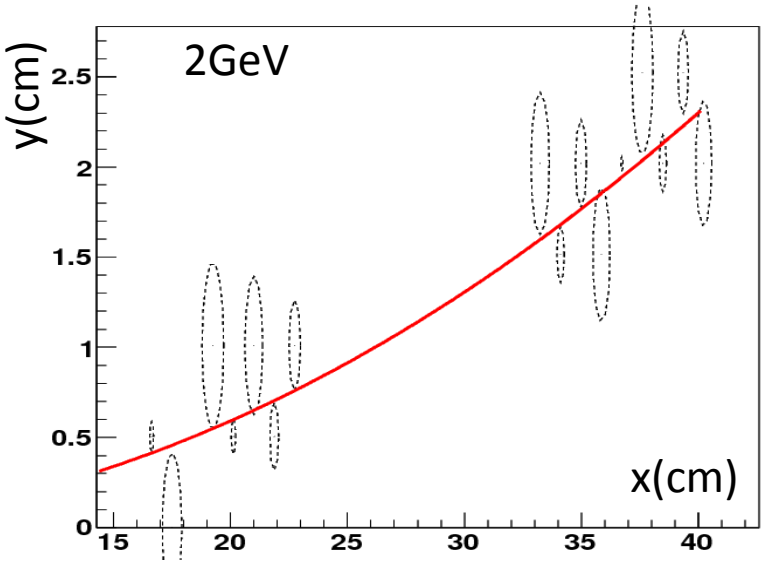
Event #2

Performance study – single track









Hardware components of the PANDA DAQ

AMC daughter board:

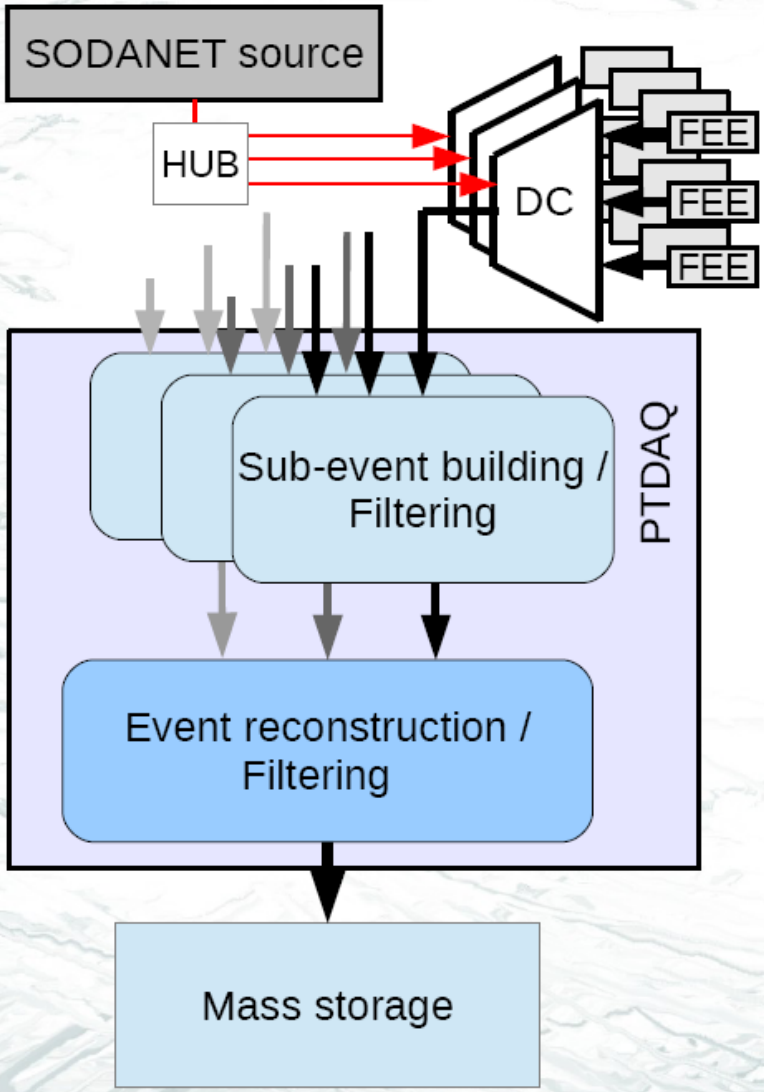
- AMC form factor
- Xilinx Virtex 5FX70T-2
- 2 x 2 GB DDR2
- 4 SFP+ interfaces
  - 6.25 Gbit optical
- 1 Gb Ethernet

ATCA based carrier board prototype

ATCA based carrier board:

- Xilinx Virtex 4FX60
- Carries up to 4 AMC cards
- Supplies direct high speed interconnection between all 4 AMCs
- Connects all 4 AMC cards via a switch FPGA to the ATCA backplane

Prototype Trigger-less DAQ



Used for testing:

- Sub detector prototypes
- Reconstruction algorithms
- Hardware

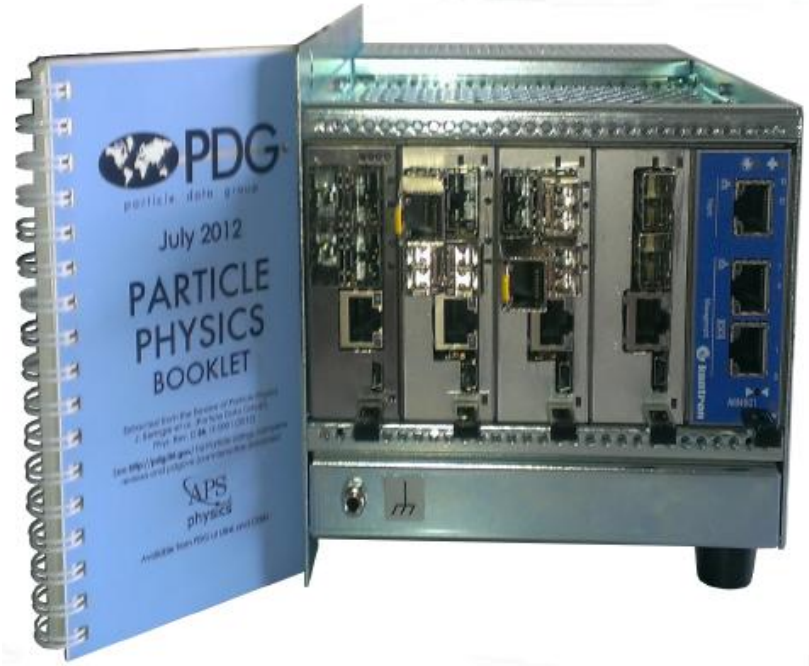
Functionality:

- Digitalized data FEE is concentrated via feature extraction and zero suppression
- Sub-event building and first filter algorithm
- Event reconstruction and second stage of filter algorithm

Differences to the $\overline{\text{P}}$ANDA DAQ:

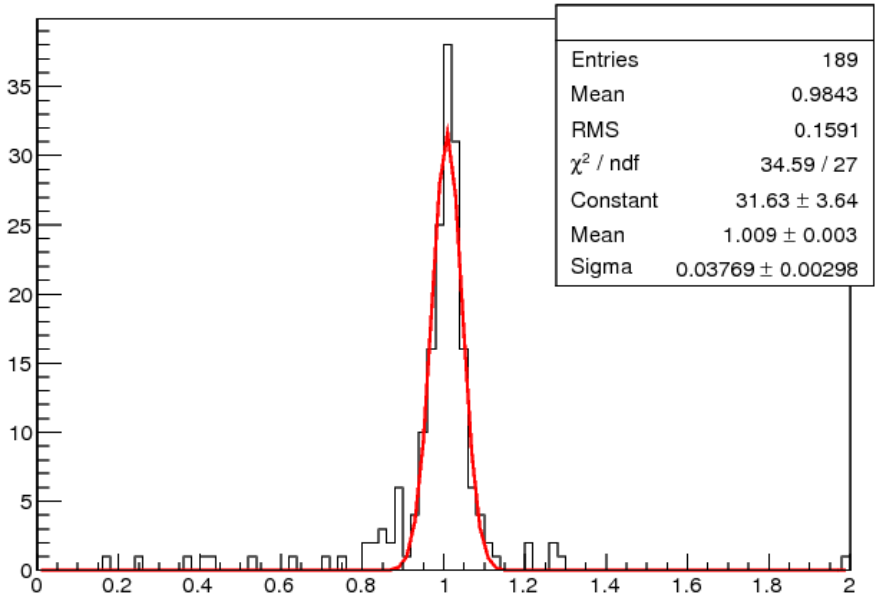
- No connection between sub-event building boards
- Less interaction / data rate

PTDAQ Setup



- Micro TCA shelf
- Up to4 x AMC daughter boards
- Up to 9 x data concentrators

Pz reconstruction



Pt = 0.5 GeV/c

Pz(input)
0.25 GeV/c : 3.7 %
0.50 GeV/c : 4.0 %
1.00 GeV/c : 4.3 %

Pt = 1.0 GeV/c

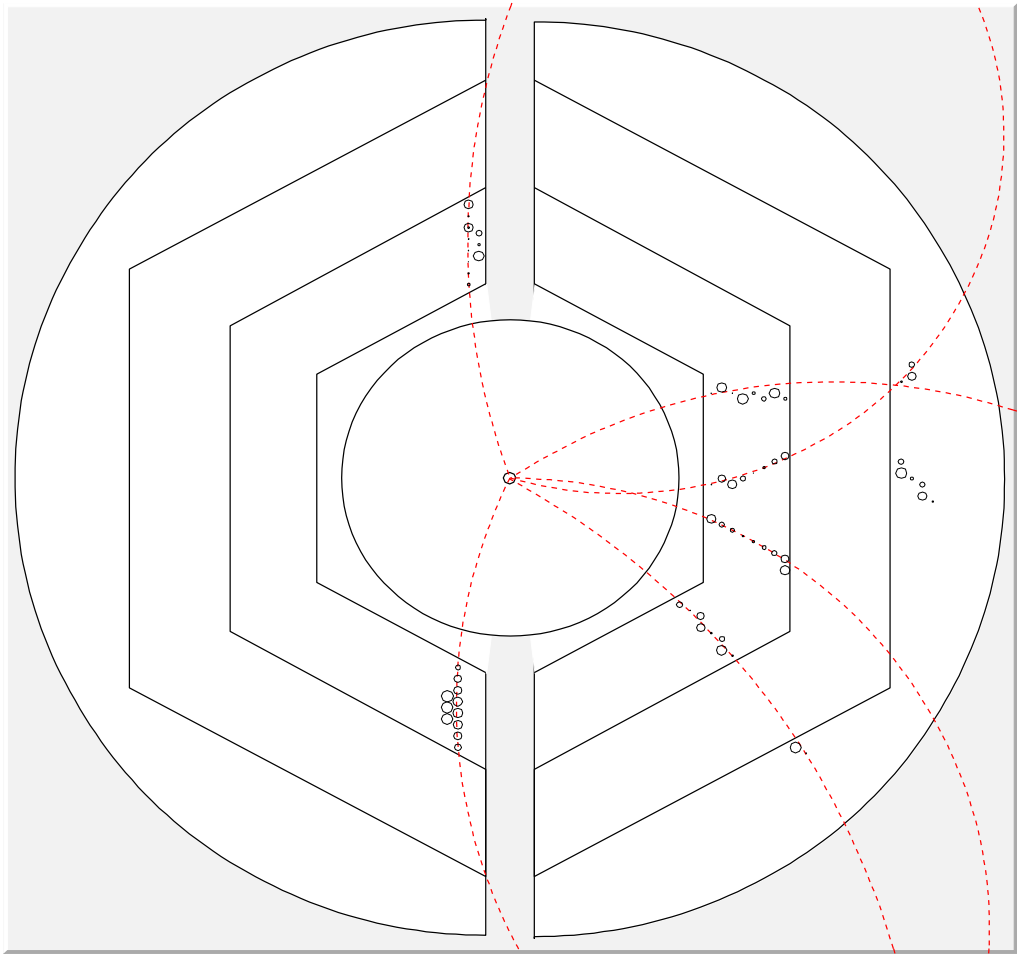
Pz(input)
0.5 GeV/c : 3.1 %
1.0 GeV/c : 3.8 %
2.0 GeV/c : 4.2 %

Pt = 2.0 GeV/c

Pz(input)
1.0 GeV/c : 3.7 %
2.0 GeV/c : 4.8 %
4.0 GeV/c : 5.2 %

Tracking at FPGA -- at low event rate

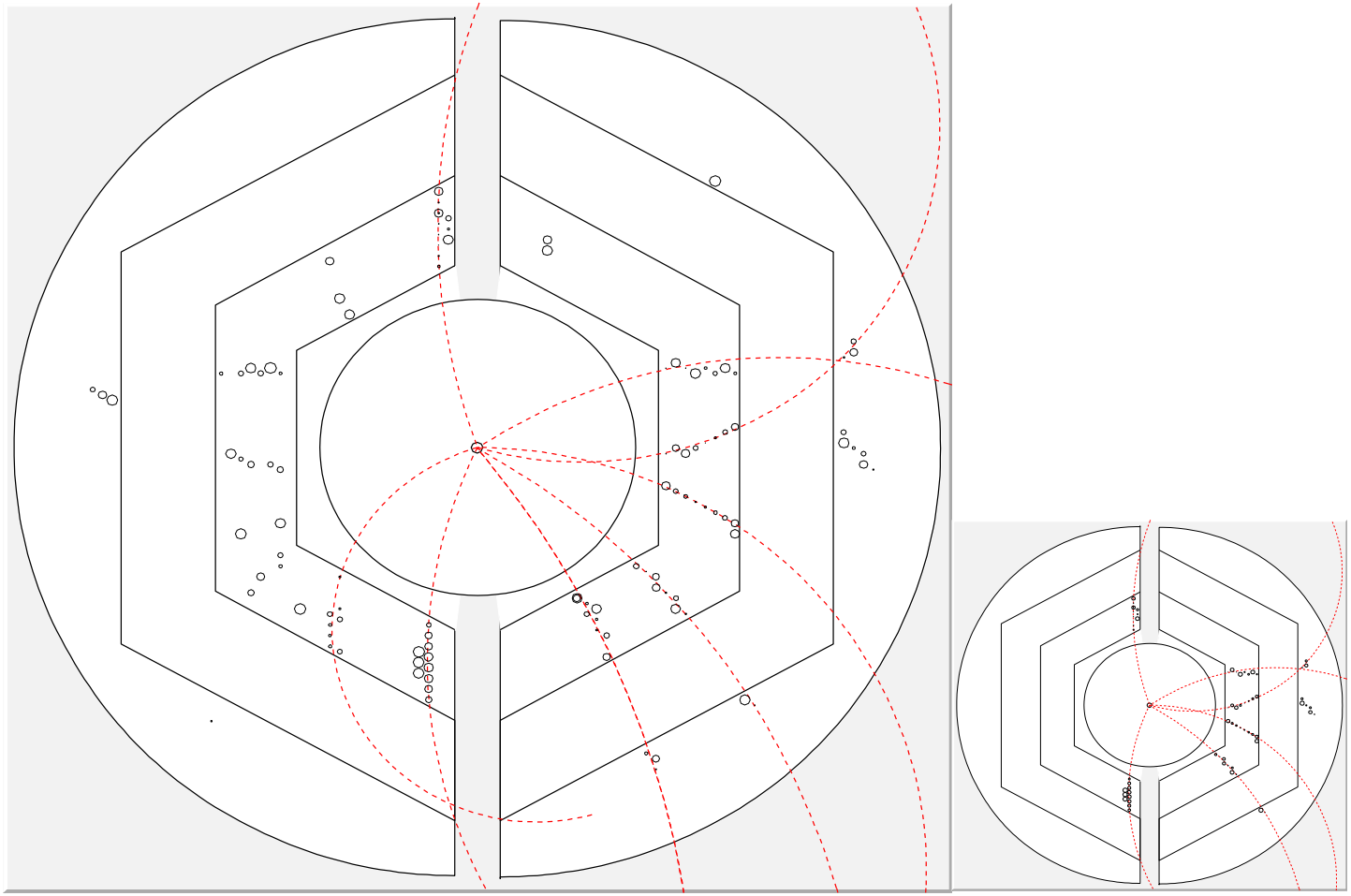
Data flow: Hit information at PC → FPGA, extract helix para. → PC



PC is used to draw hits and helix in the plot. The helix parameters come from FPGA.

Tracking at FPGA -- 20 MHz

Data flow: Hit information at PC → FPGA, extract helix para. → PC



PC is used to draw hits and helix in the plot. The helix parameters come from FPGA.