FPGA Helix Tracking Algorithm with STT

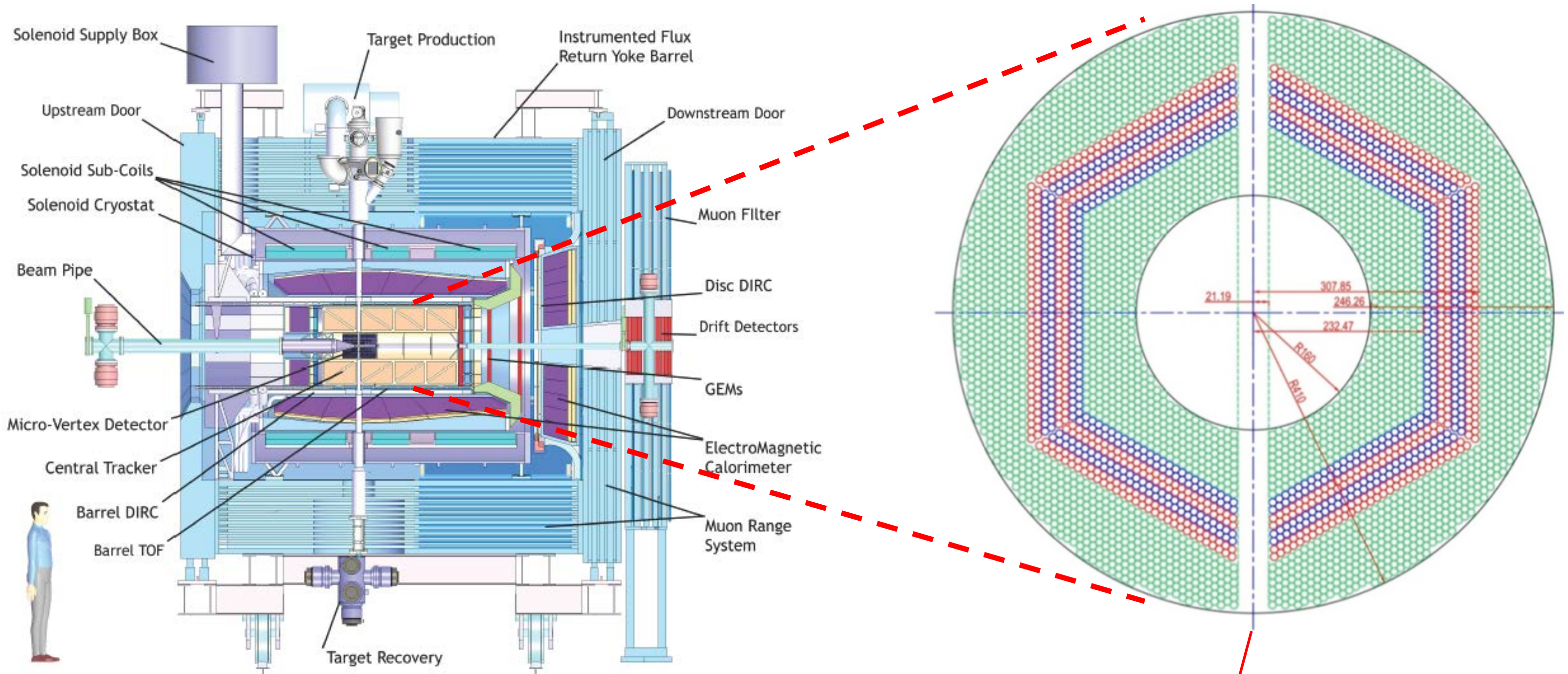
Yutie Liang, Thomas Geßler,
Jens Sören Lange, Wolfgang Kühn,

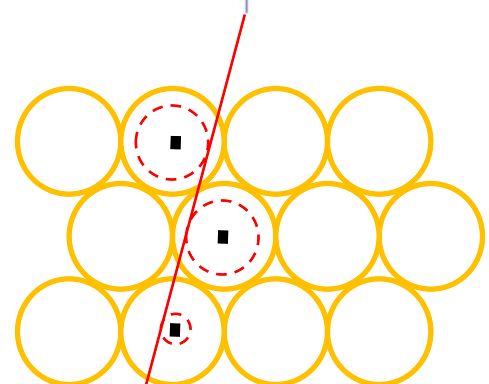
II. Physikalisches Institut, JUSTUS-LIEBIG-UNIVERSITÄT GIESSEN
Institute of Modern Physics, Chinese Academy of Sciences
19.09.2018

Outline

1. Introduction
2. Tracking Algorithm
3. T0 extraction from tracking
4. Tracking at FPGA
5. Summary and Outlook

Introduction to the PANDA STT

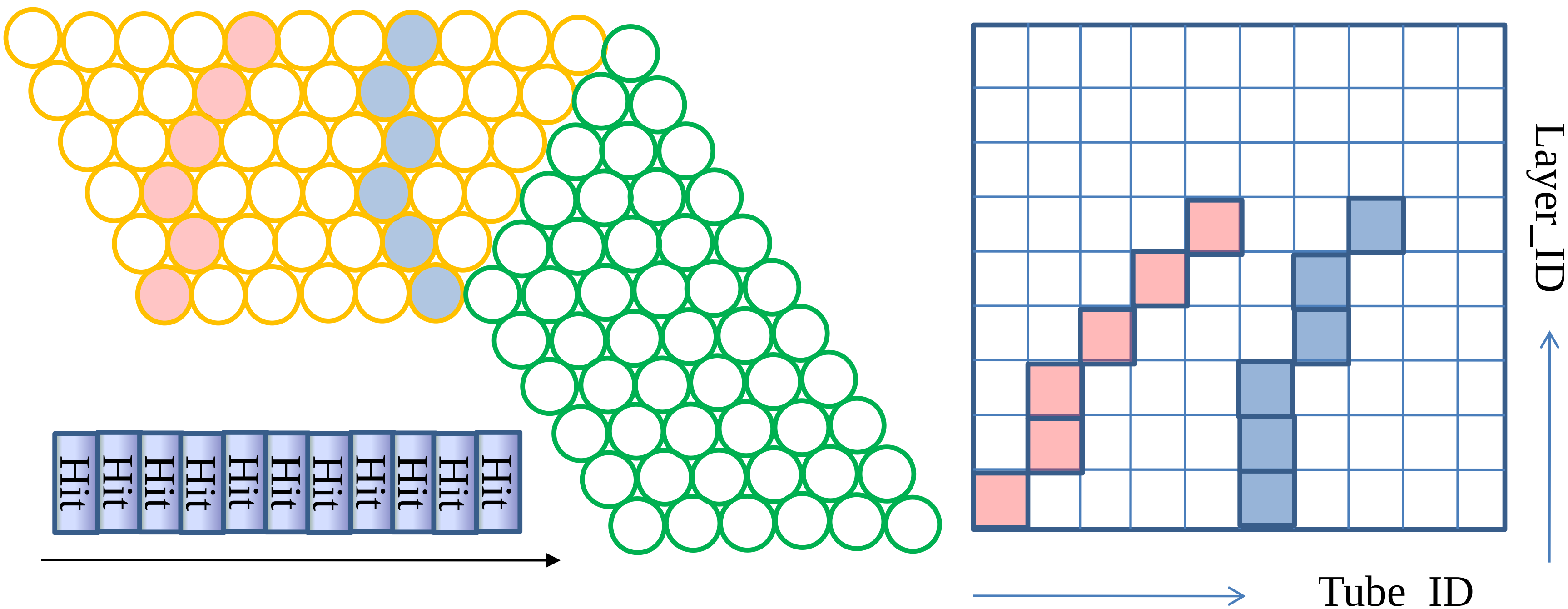




- ➢4636 Straw tubes
- ➢23-27 planar layers
- • 15-19 axial layers (green) in beam direction
- • 4 stereo double-layers for 3D reconstruction, with ±2.89 skew angle (blue/red)

From STT : Wire position + drift time

Tracking Algorithm -- Road Finding



Hit: Seg_ID (3 bits) + LayerID (5 bits) + Tube_ID (6 bits) + Arrival time

1: Start from inner layer

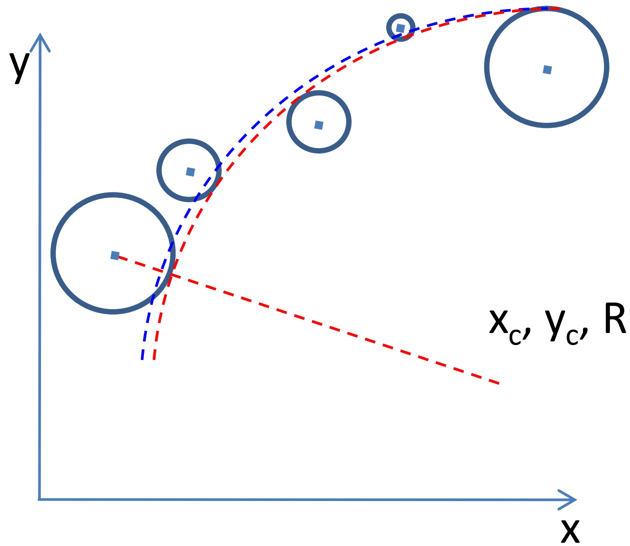
2: Attach neighbour hit to tracklet layer by layer

✓ 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.

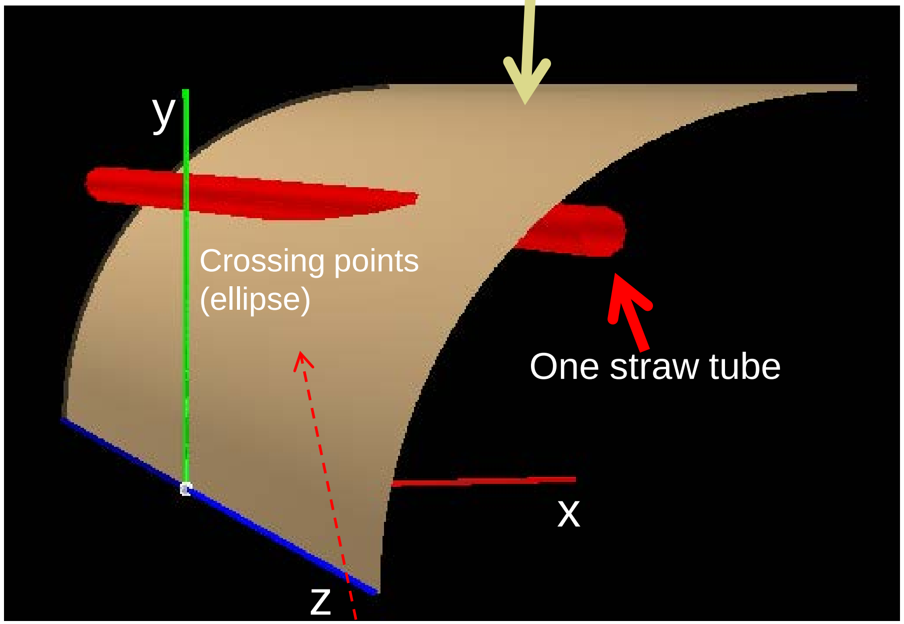
$$\chi^2 = 1/n \times \sum_i \frac{(x_i{}^2 + ax_i + y_i{}^2 + by_i)/2r^2}{d_i{}^2}$$

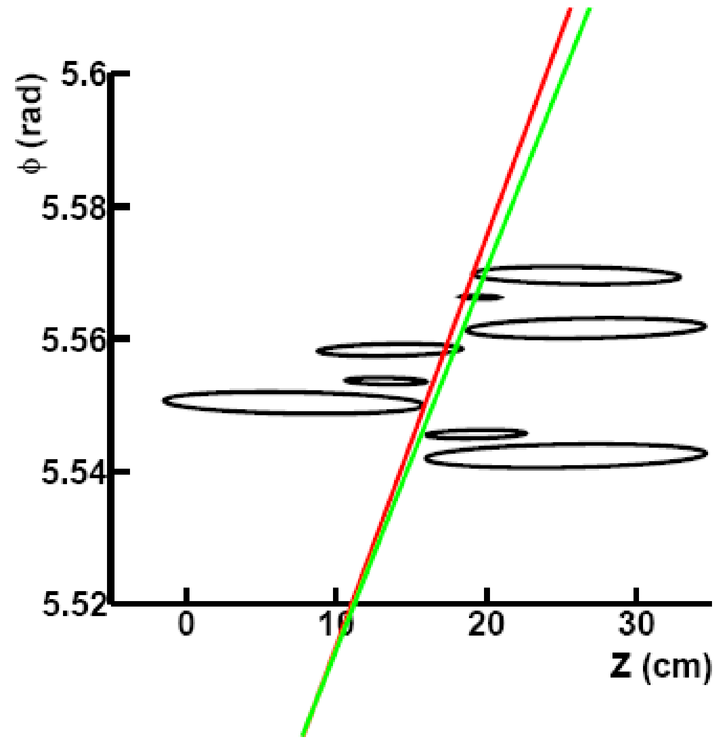
Pz reconstruction

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

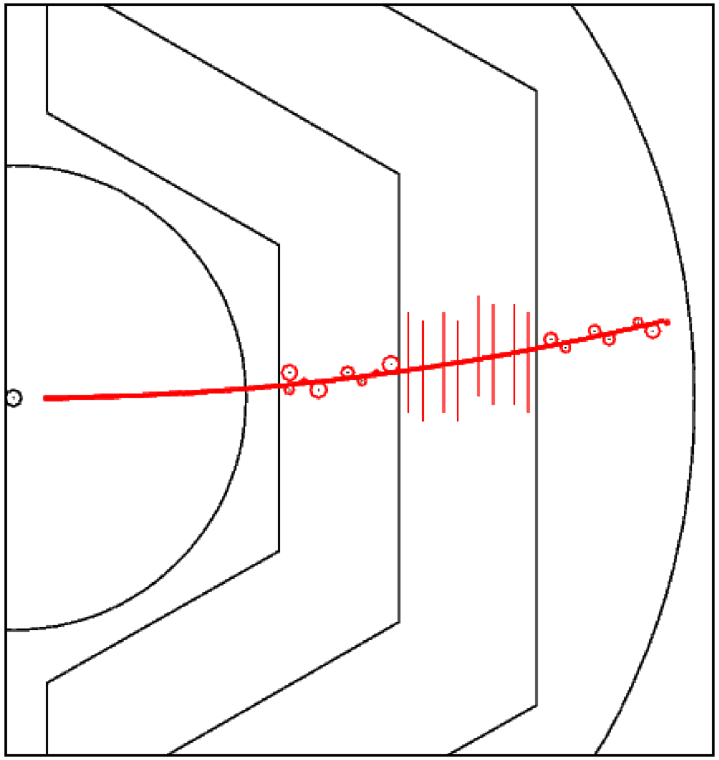


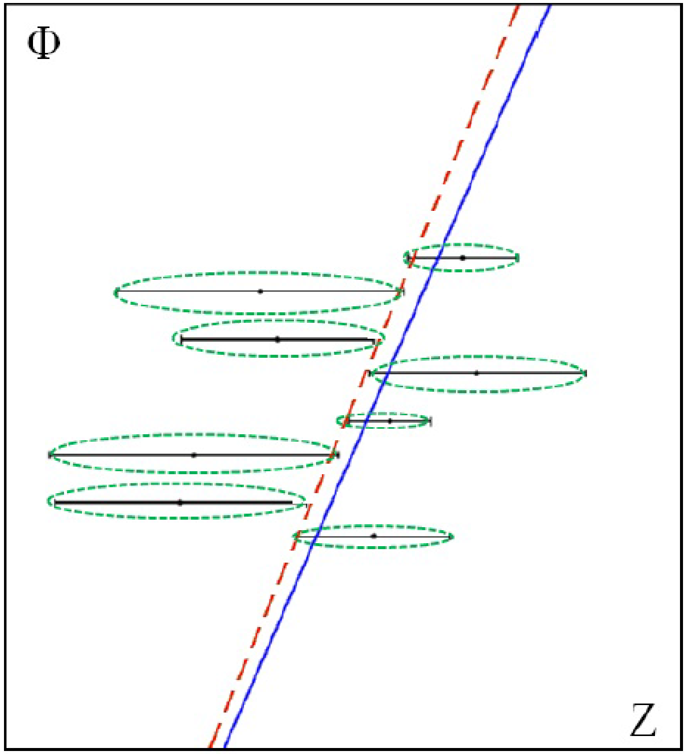


Track need to be tangent to the crossing ellipse.

Gianluigi Boca and Panda Collaboration, Panda STT TDR, 2012

Pz reconstruction





Known : z_i , Φ_i , d_i

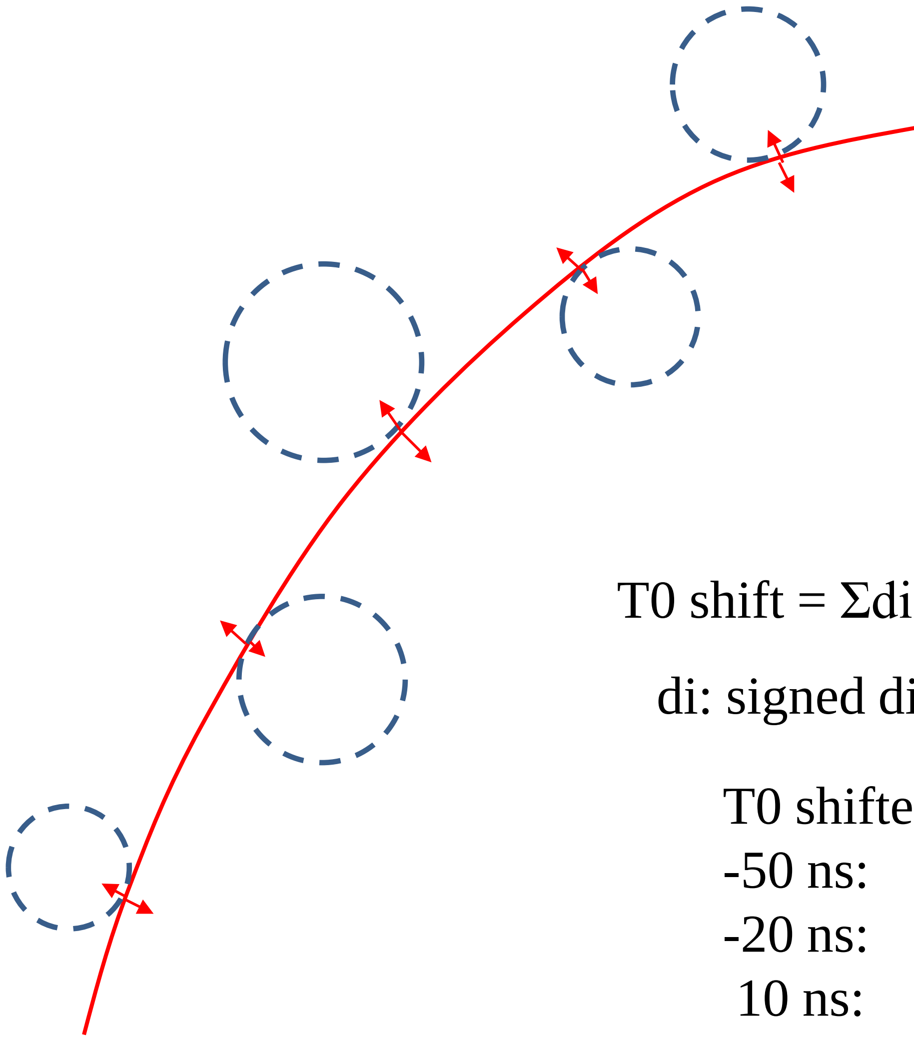
Question: To determine a line,

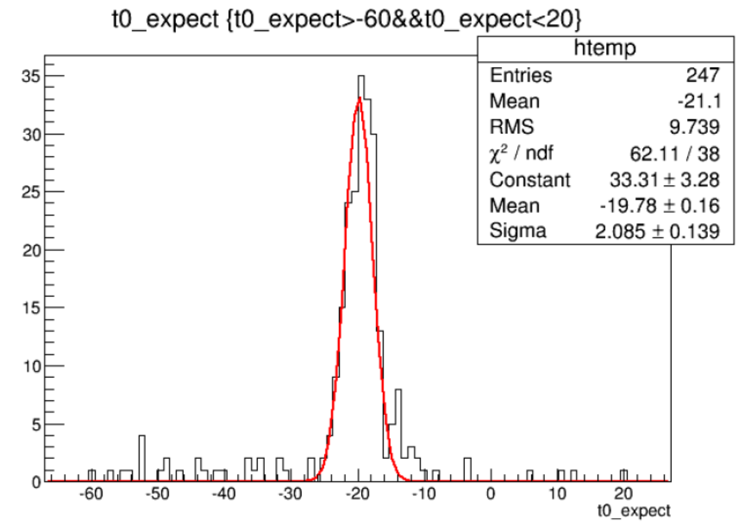
$\Phi + kz + \Phi_0 = 0$

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

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

Extract T_0 from Tracking





T0 shift = Σdi /N / const (N: number of hits

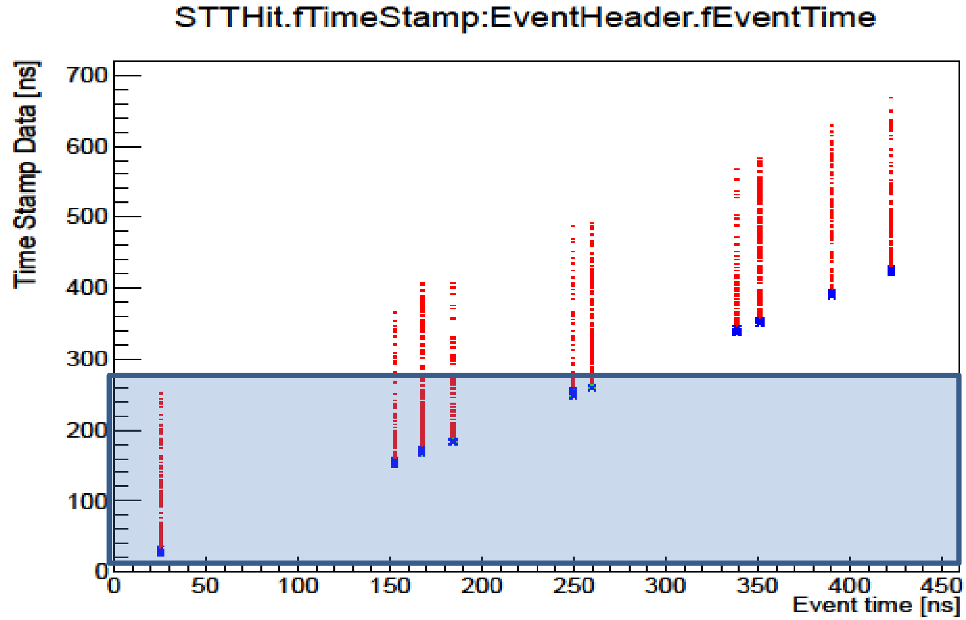
di: signed distance of circle to track)

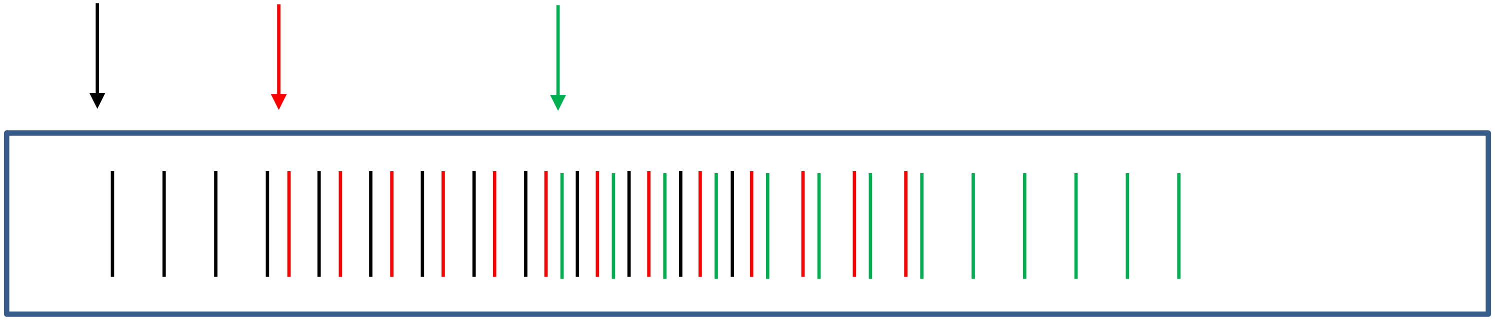
T0 shifted by:	Extracted T0:
-50 ns:	-47.0 ± 4.0 ns
-20 ns:	-19.8 ± 2.1 ns
10 ns:	9.4 ± 2.5 ns
20 ns:	19.0 ± 2.3 ns

Event Structure -- pile-up

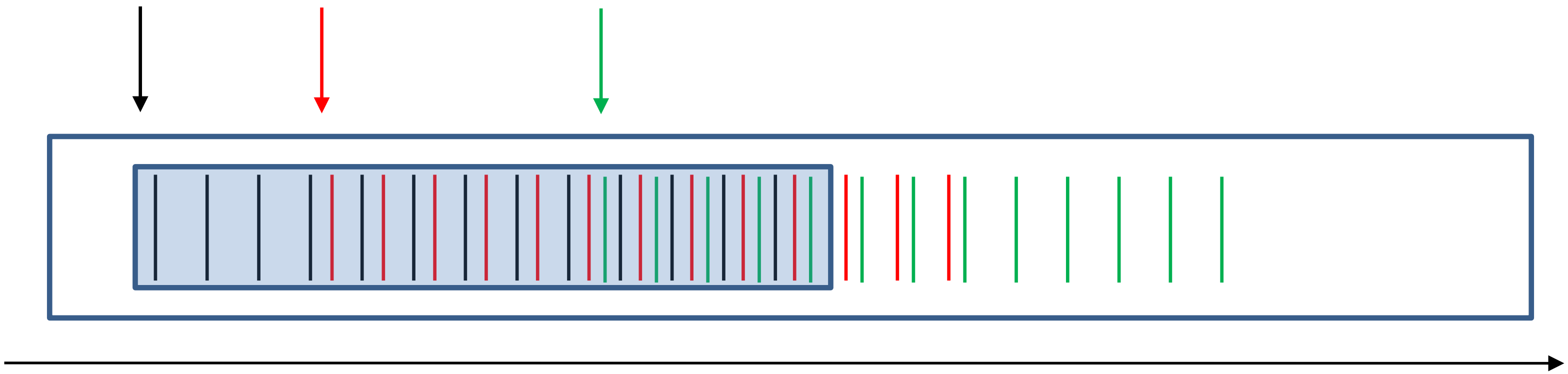
- 20 MHz
- 1600 ns revolution time
- 400 ns gap
- T_{Mean} between 2 events: 50 ns
- Drift time: ~200 ns

Event pile-up





Tracking Strategy with experimental data -- 20 MHz



Hits are sorted according to their arrival time.

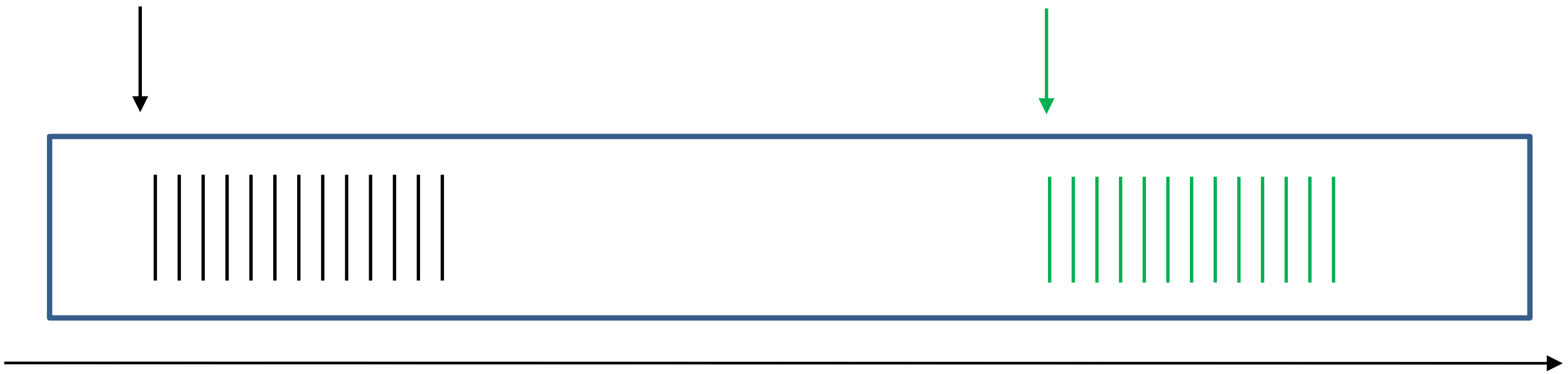
In case of T_0 provided:

Start from T_0 → Collect hits in 220 ns window →
Run tracking algorithm → Assign track to right event

In case of T_0 not provided:

Do tracking and T_0 extraction at the same time

Tracking Strategy with experimental data -- Phase I



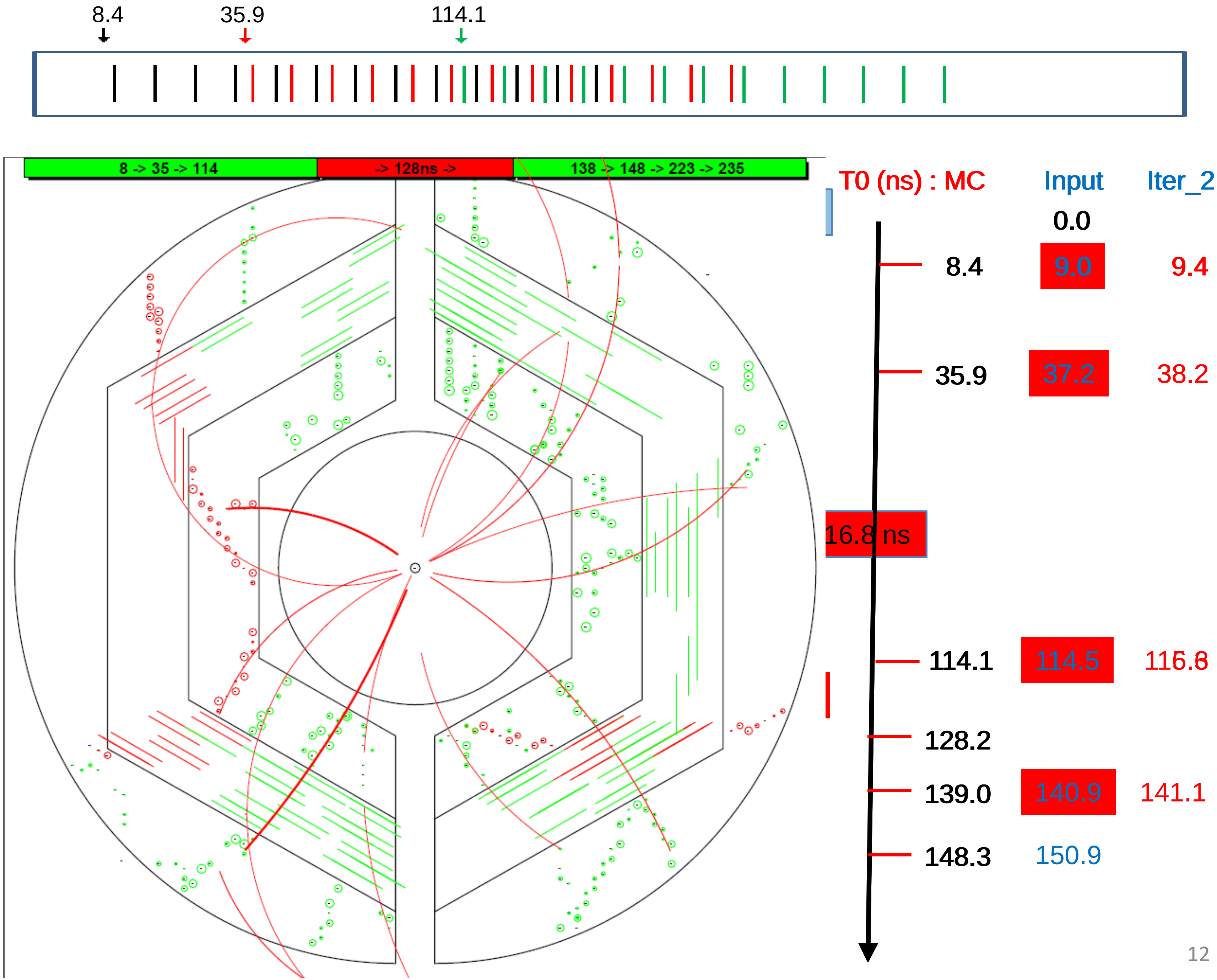
Hits are sorted according to their arrival time.

In case of T_0 provided:

Start from T_0 → Collect hits in 220 ns window →
Run tracking algorithm → Assign track to right event

In case of T_0 not provided:

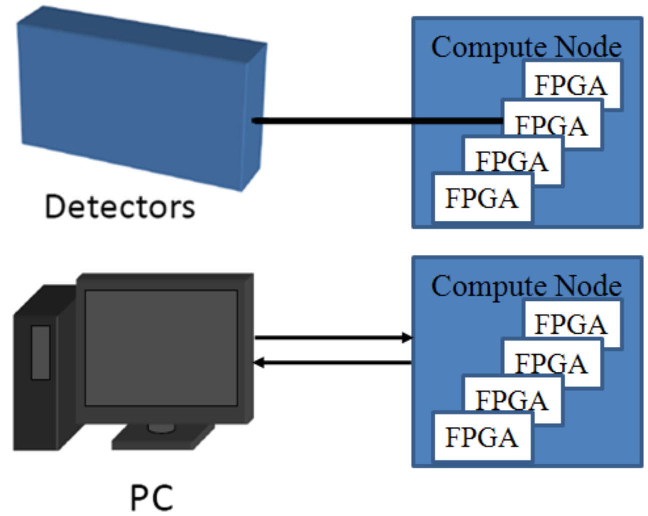
Take the earliest time of a segment of hit as T_0 . → do one more iteration to optimize.

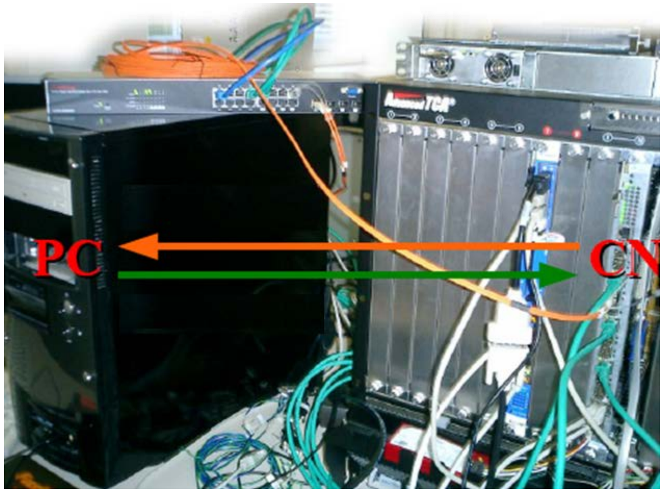

8.4
35.9
114.1
8 -> 35 -> 114
-> 128ns ->
138 -> 148 -> 223 -> 235
T0 (ns) : MC
Input
Iter_2
0.0
8.4
9.0
9.4
35.9
37.2
38.2
16.8 ns
114.1
114.5
116.8
128.2
139.0
140.9
141.1
148.3
150.9

Setup and Test

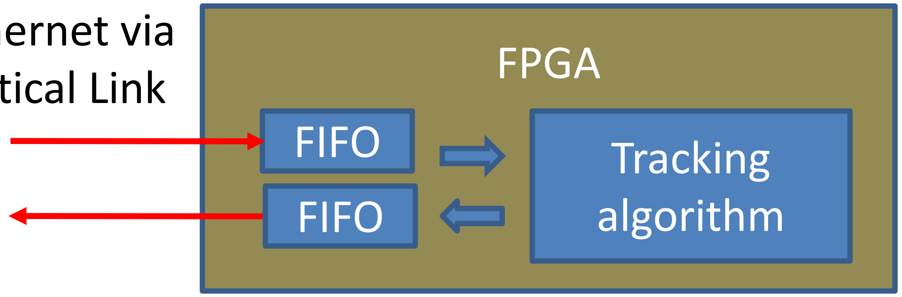
PC as data source and receiver.

- Ethernet.
- Optical link

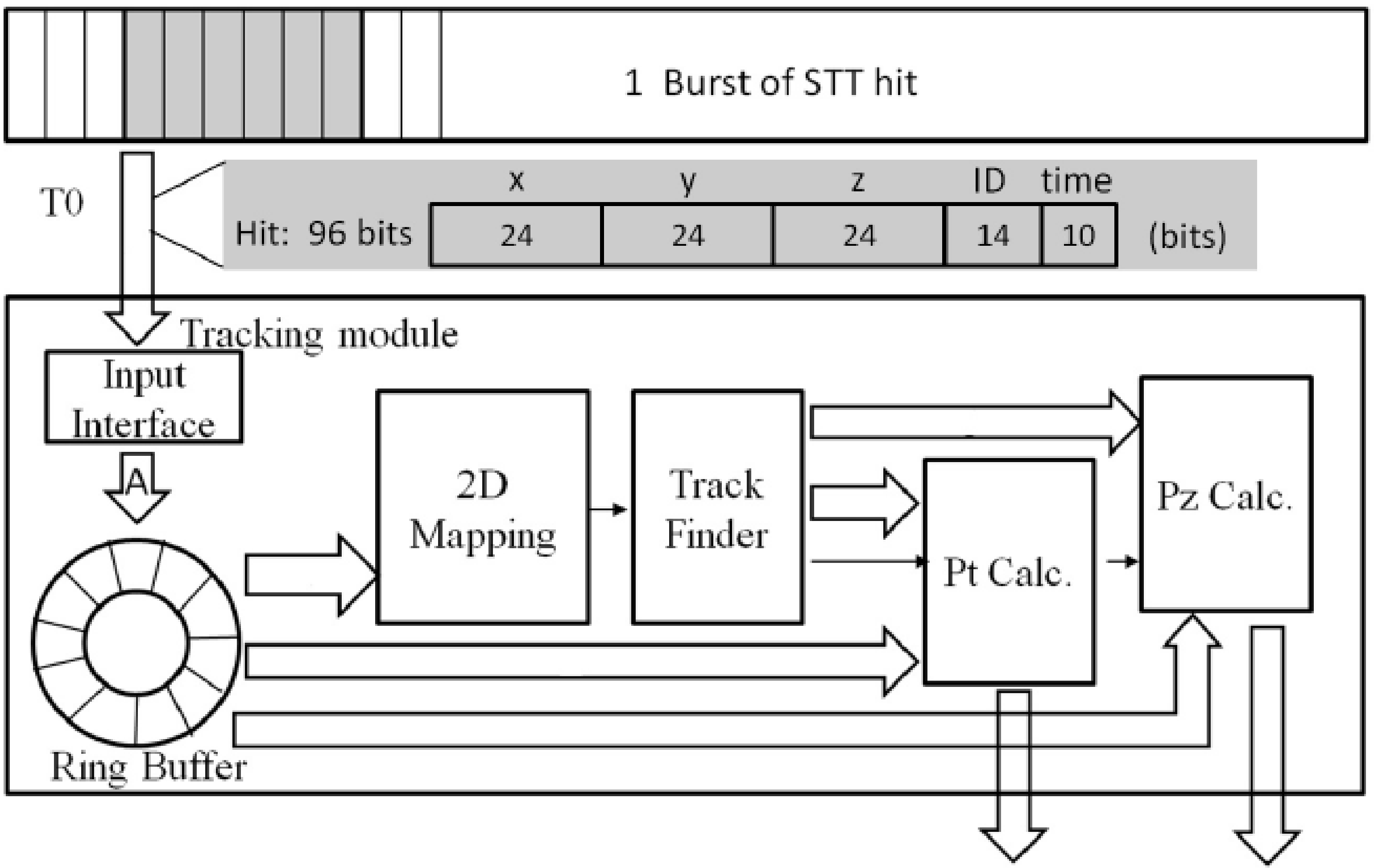




Ethernet via
Optical Link



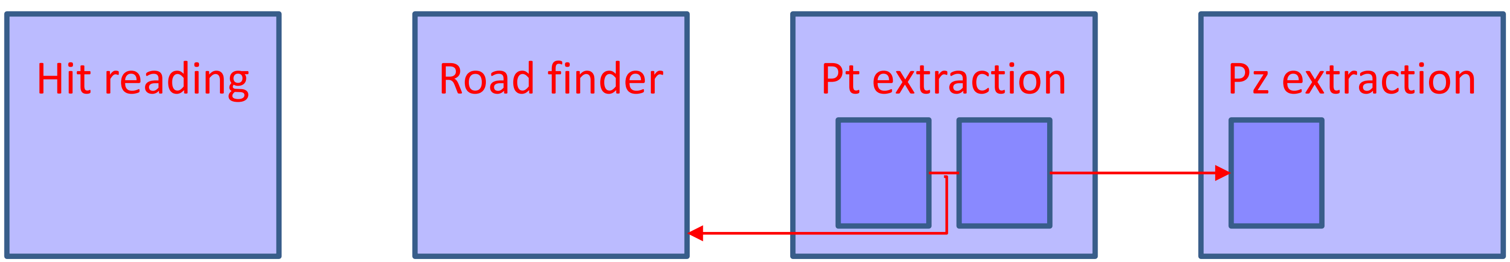
Block diagram of algorithm in VHDL



Performance at FPGA

Device Utilization Summary			
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice Flip Flops	25,022	50,560	49%
DCM autocalibration logic	14	25,022	1%
Number of 4 input LUTs	33,120	50,560	65%
DCM autocalibration logic	8	33,120	1%
Number of occupied Slices	21,563	25,280	85%
Number of FIFO16/RAMB16s	148	232	63%
Number used as RAMB16s	148		
Number of DSP48s	124	128	96%

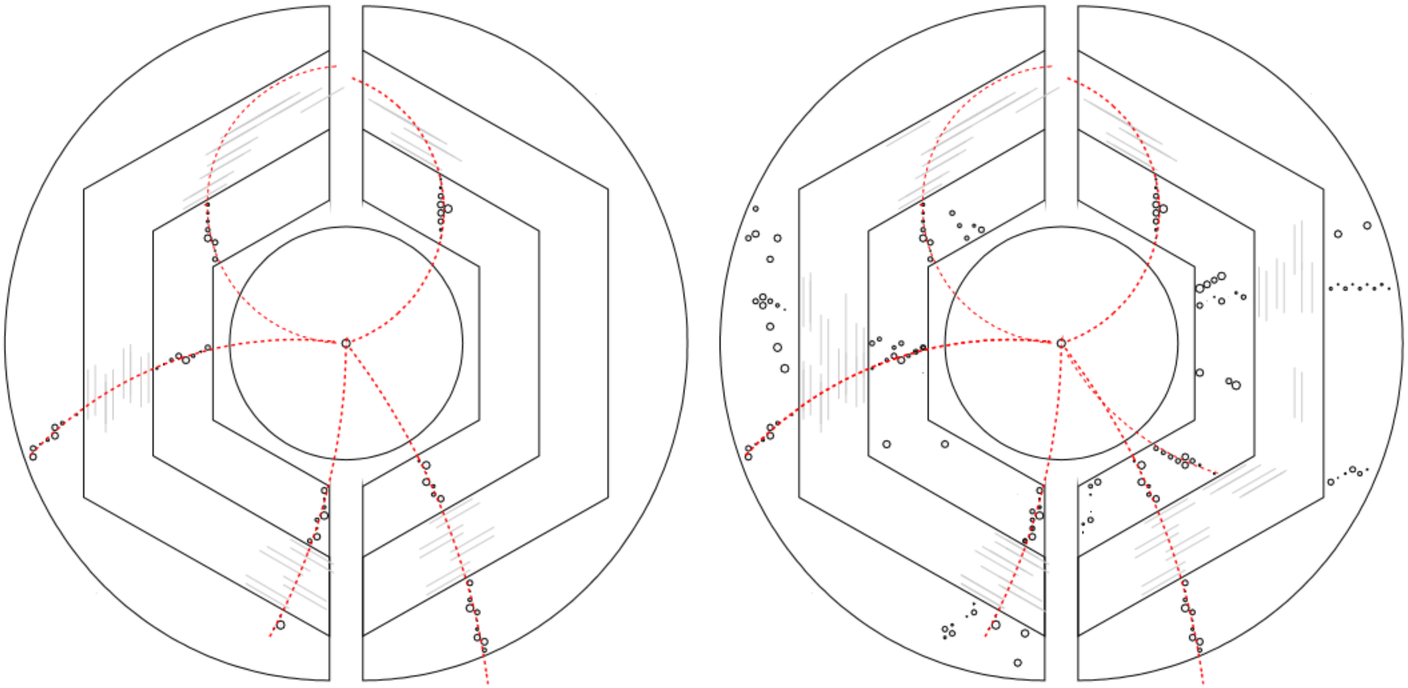
For one event with 100 hits (6 tracks): 7 μs



(100 cc) (50 cc X 6) (120 cc X 2 X 6) (60 cc X 2 X 6)

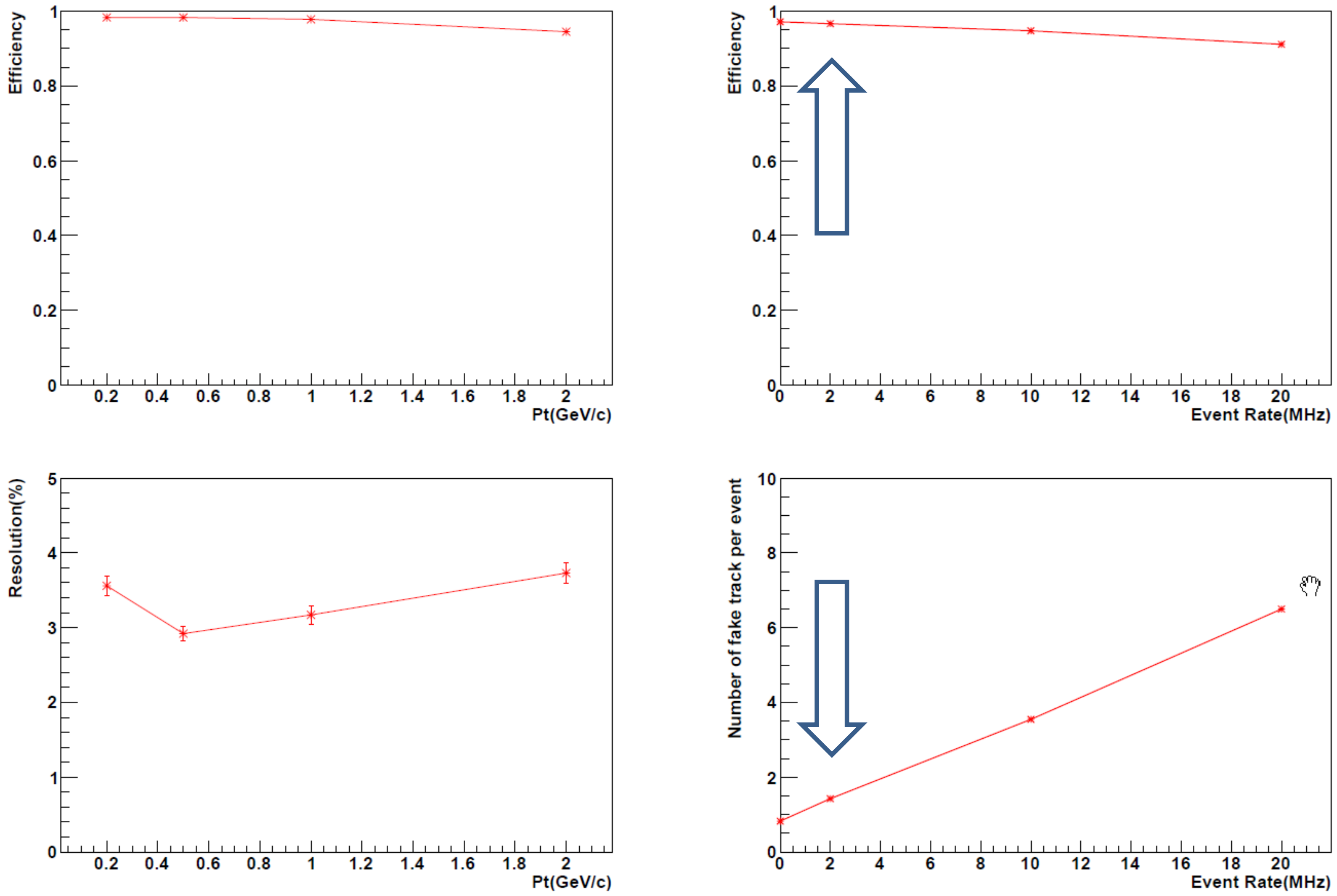
Tracking at FPGA

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

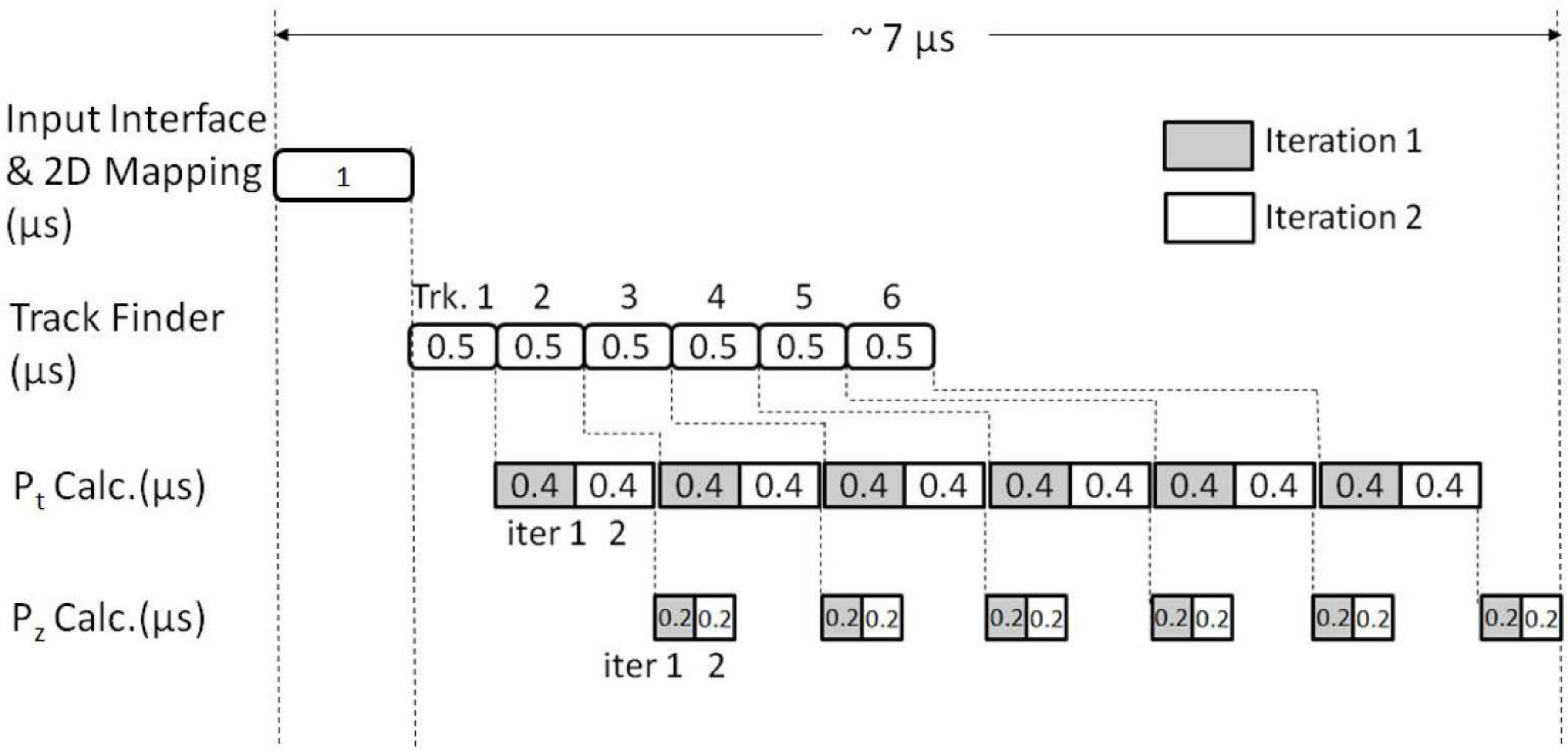


PC is used to draw hits and helix in the plot. The helix parameters come from FPGA.

Performance test



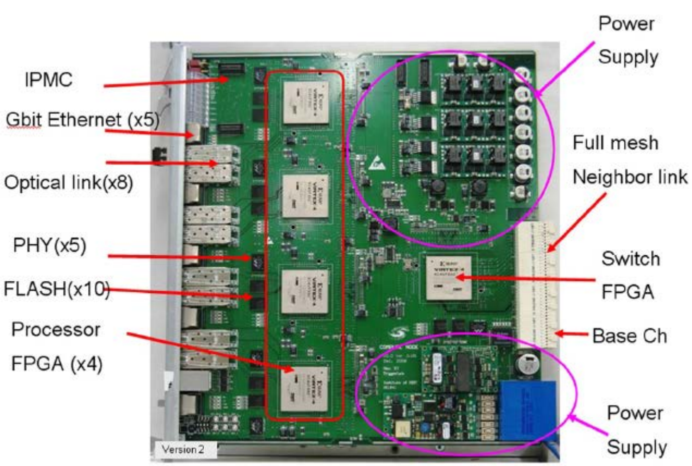
The Latency



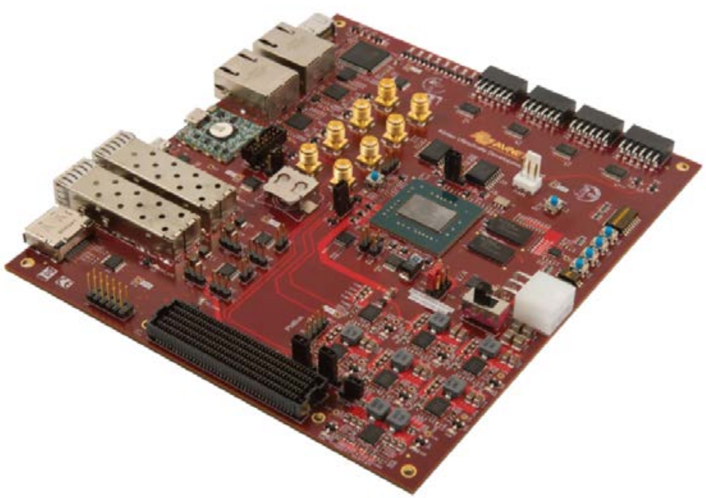
~ 7 μs
Input Interface & 2D Mapping (μs)
1
Iteration 1
Iteration 2
Track Finder (μs)
Trk. 1 2 3 4 5 6
0.5 0.5 0.5 0.5 0.5 0.5
Pt Calc.(μs)
0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4
iter 1 2
Pz Calc.(μs)
0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2
iter 1 2

Summary and Outlook

- σ_pt : ~ 3.2% σ_pz : ~ 4.2% .
- 7 μs/event (6 tracks)
- Tracking strategy in case of W/O T0.



Sitcp: Ready



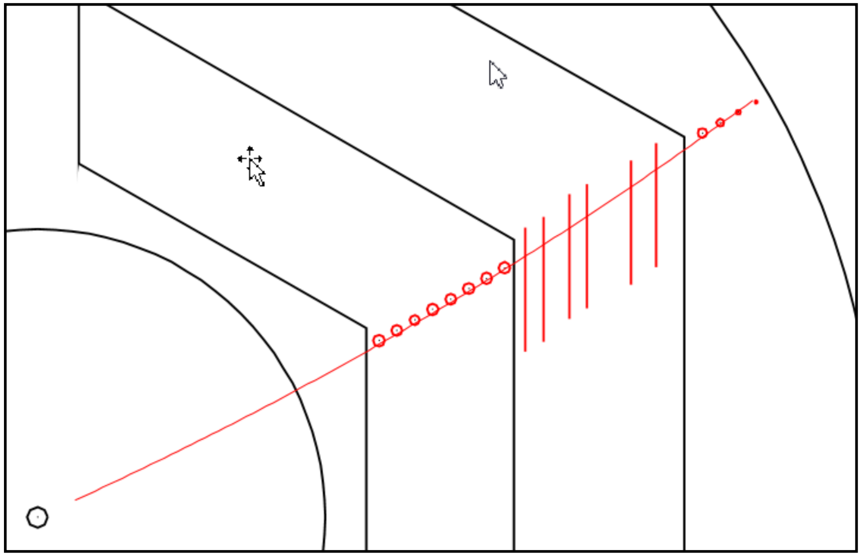
Thank you

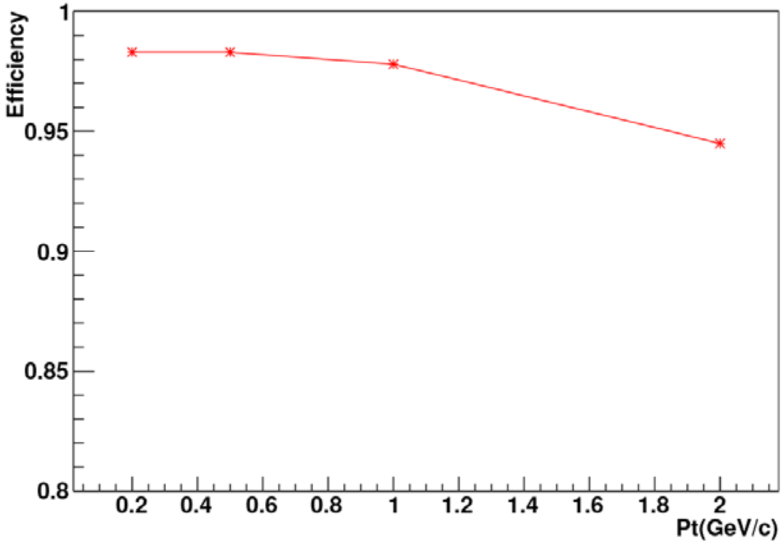
Performance test

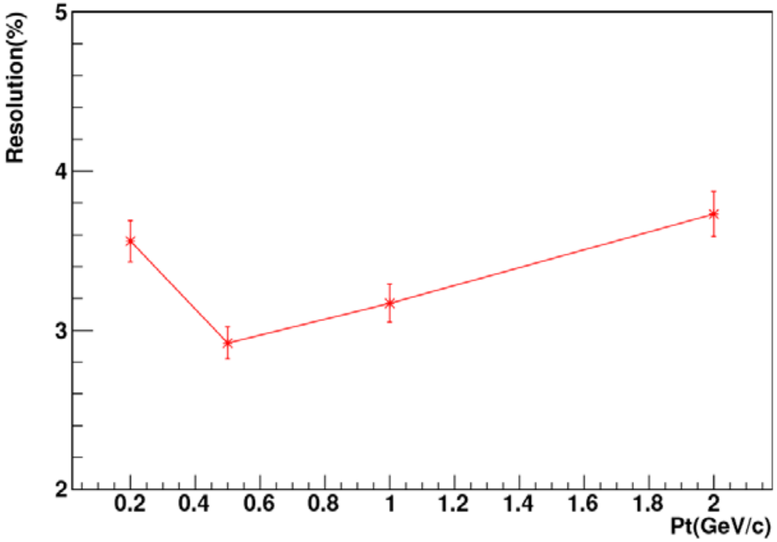
$$\epsilon = N_{rec}/N_{gen}$$

N_gen: number of generated tracks
with at least 3 hits in STT

|P_rec - P_gen| < 6 sigma

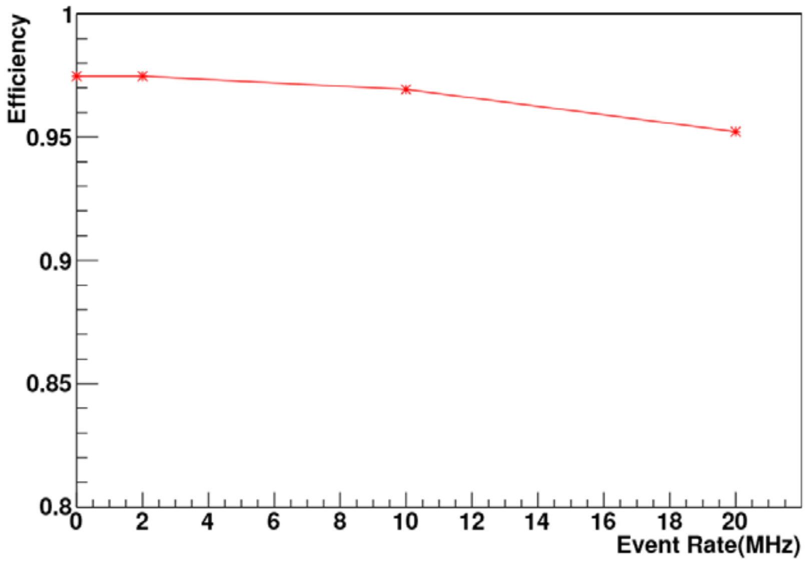


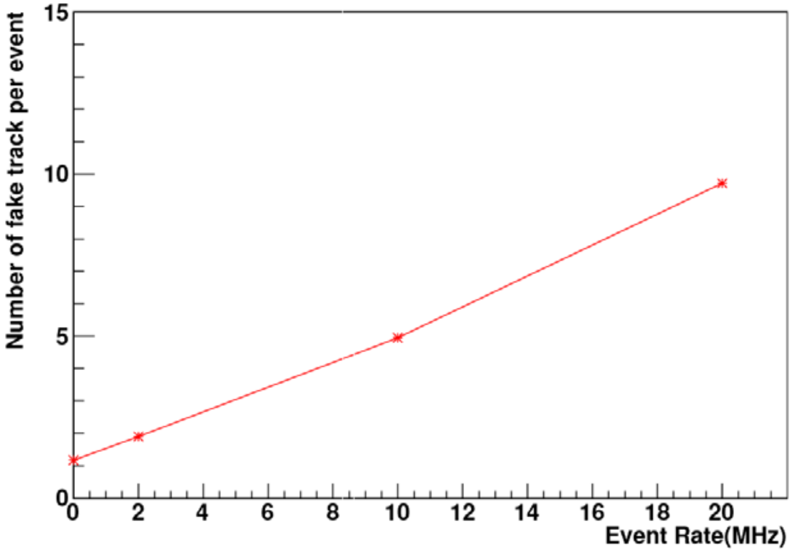




Performance at different event rate

Number of fake tracks per event increases at high event rate.

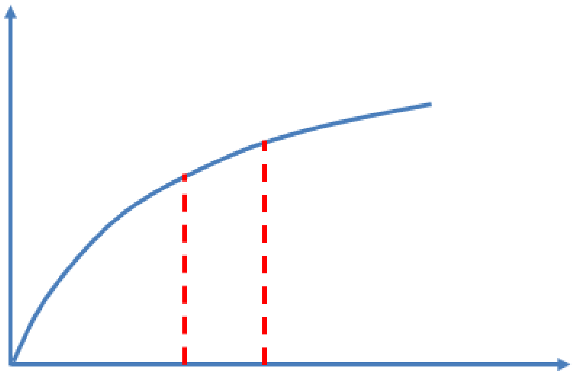




Square Root Operation

384 bins in the range of 1 ~ 4, error ~ 0.4%

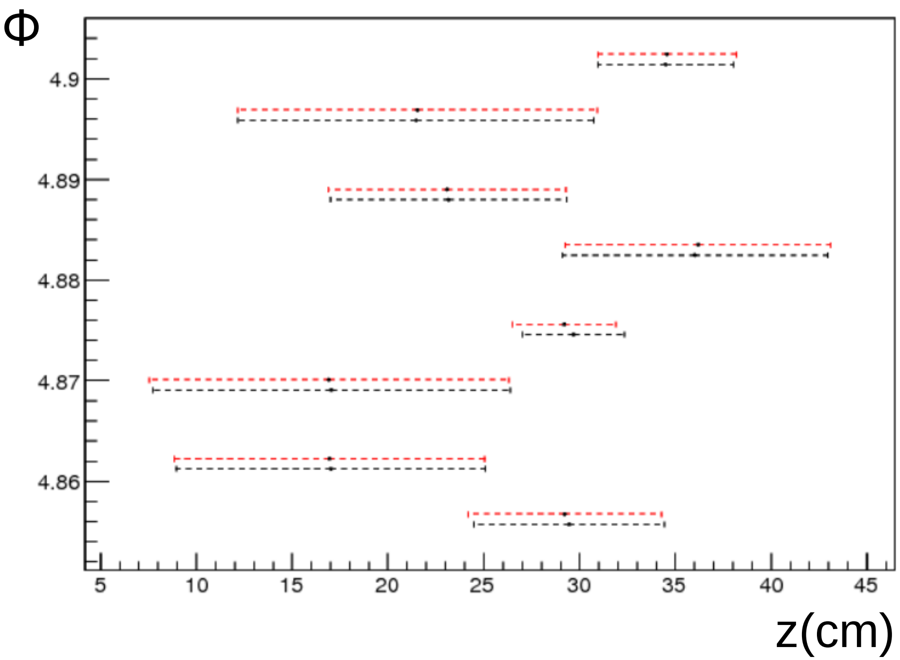
Linear extrapolation, precision improves



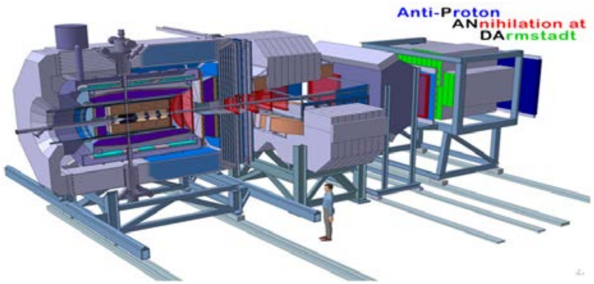
	C++	VHDL	
Xc:	0.7286	0.7278	
Yc:	161.153	161.138	
R:	161.155	161.167	(0.01%)

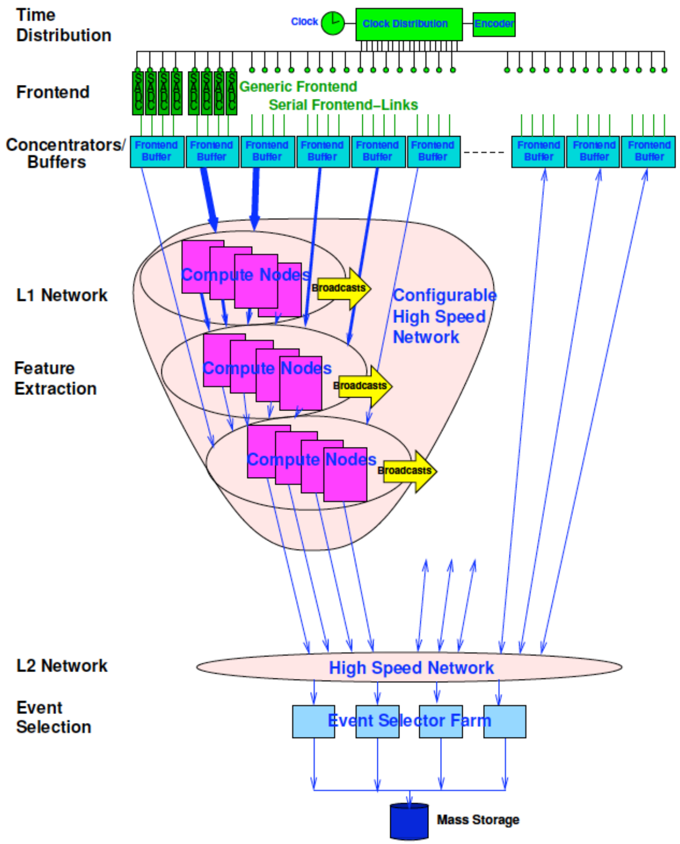
Zi:

C++	VHDL
29.4 ±5.0	29.23 ±5.1
17.0 ±8.1	16.93 ±8.1



The Structure of PANDA TDAQ



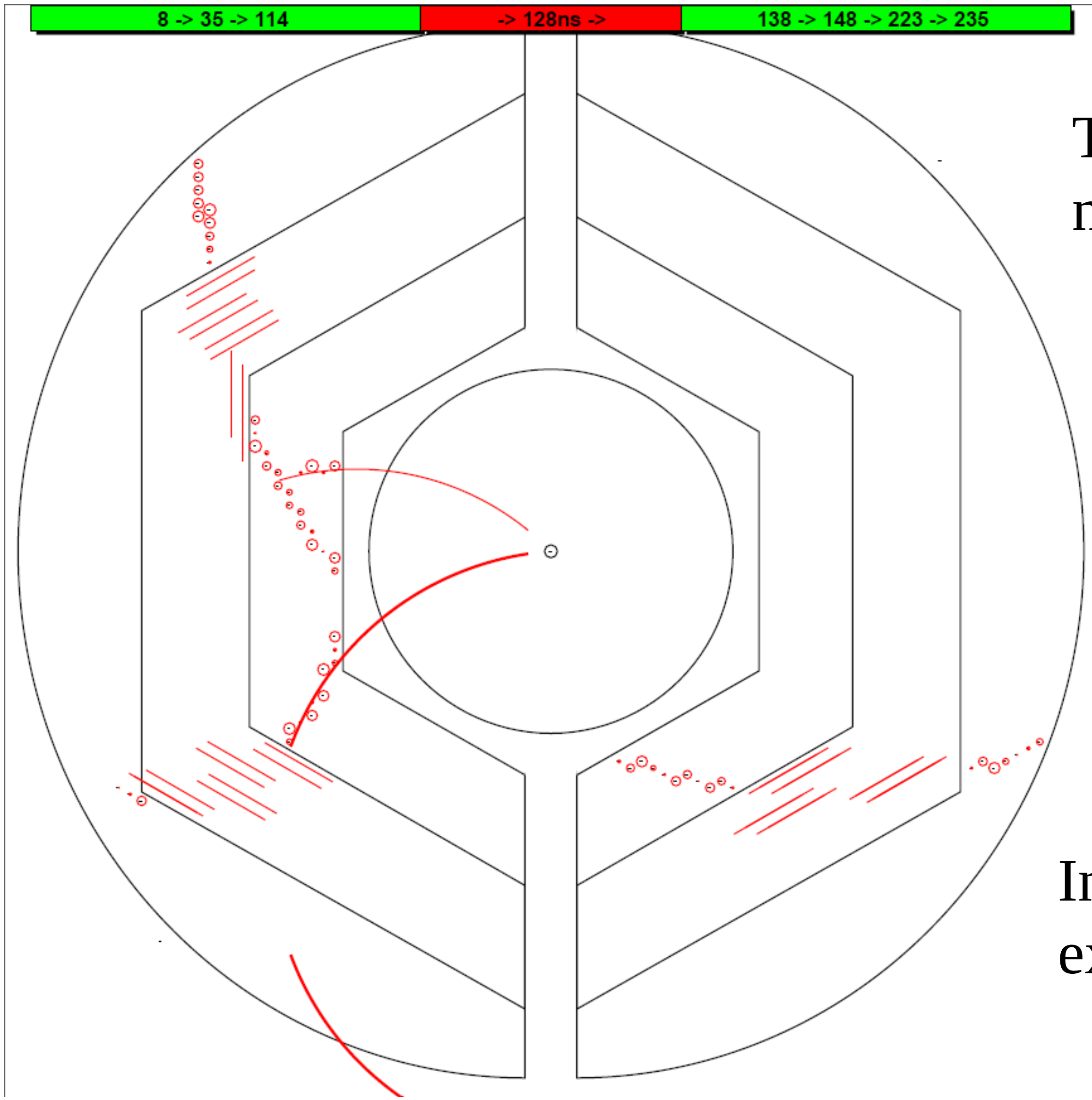


- **Time Distribution System** – provide clock for hit timestamps
- **Concentrators/Buffers** – buffering and on the fly data flow manipulation
- **L1 Compute Nodes** – mark hits which might belong to the same event (time slice)
- **L2 Feature Extraction Nodes** – combine detector information to extract physical signatures (momentum, …)
- **L3 Event Selection Nodes** – event selection based on a complete reconstruction (PID, vertex, invariant mass, …)

Prototype Trigger-less Data Acquisition

by Milan Wagner

The Missing Event at Event-based simulation

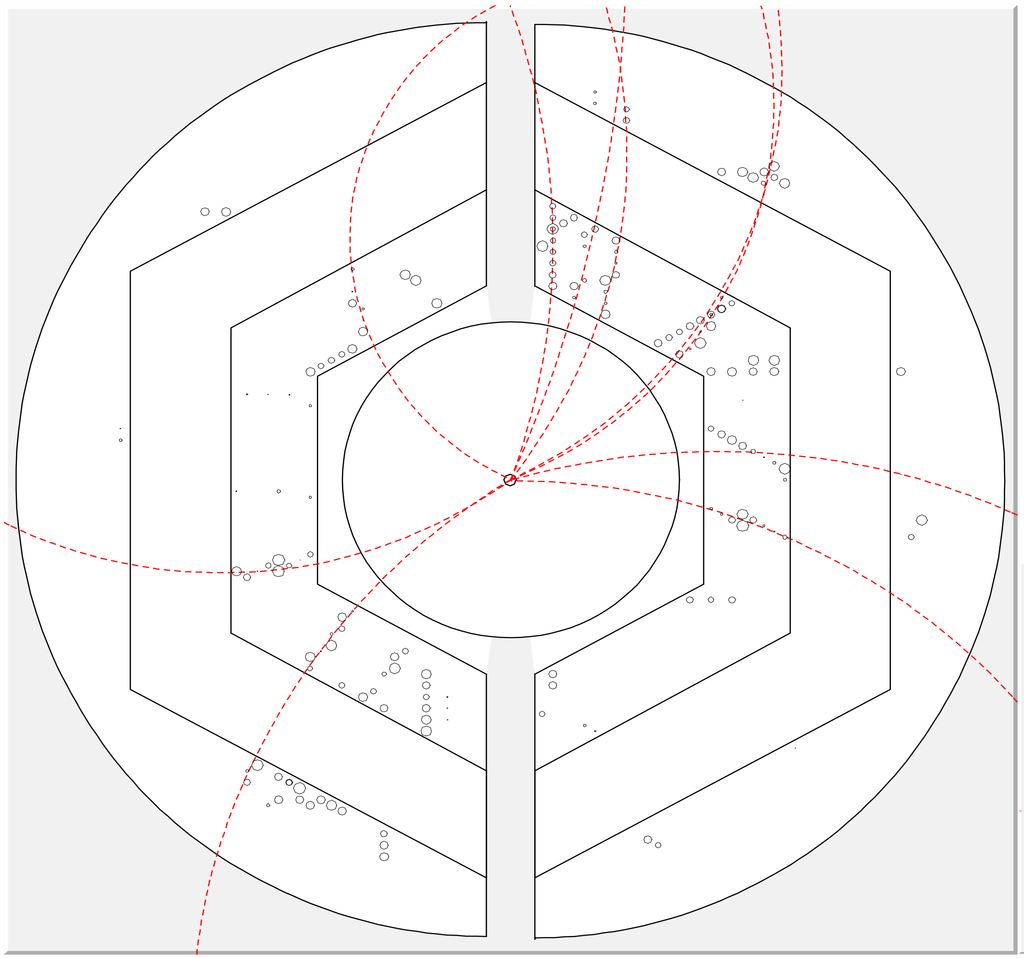


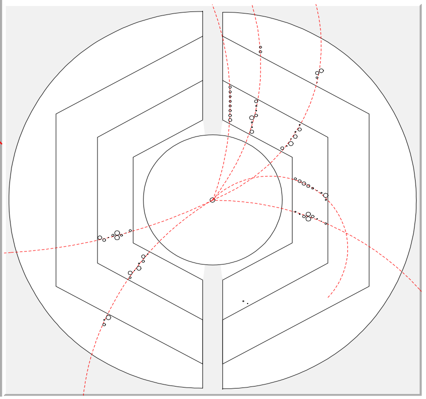
Tracks in this event are not well reconstructed.

In this case, it is hard to extract a T_0.

Tracking at FPGA -- 20 MHz

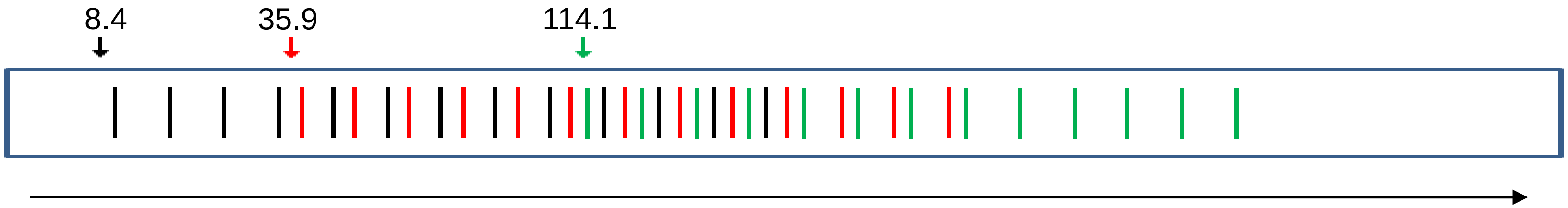
Data flow: Hit information at PC → FPGA, extract helix para. → PC





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

Tracking Strategy



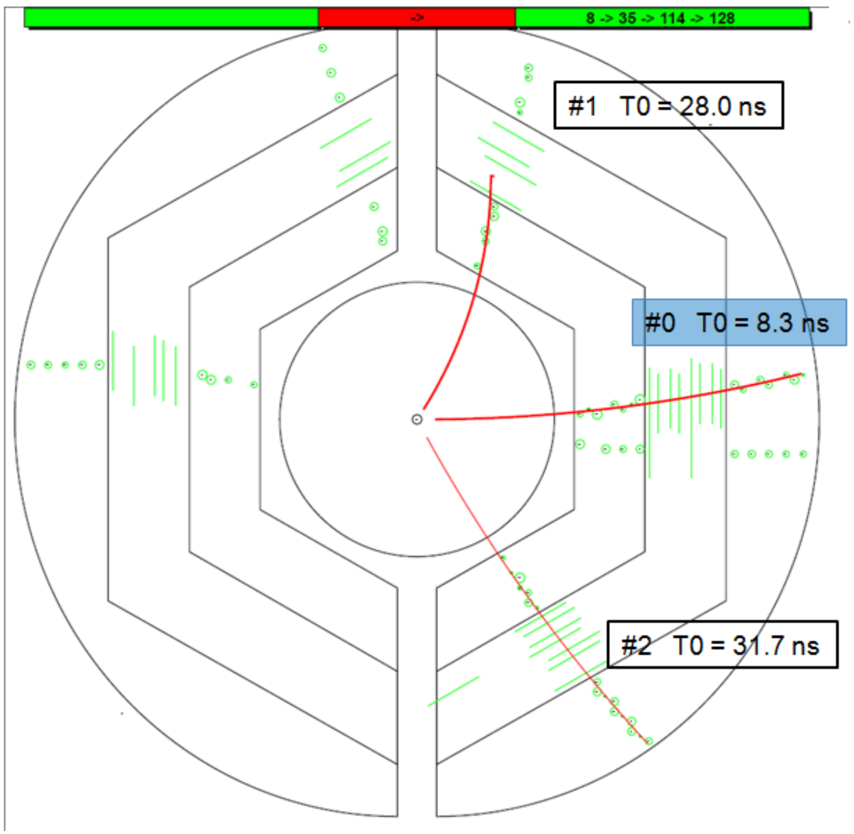
→ 1) Start from 0

→ 2) hits in 220 ns window

→ 3) run tracking algorithm

→ 4) extract T_0 of next event

→ 5) go back to 2)



Two time-based simulations:

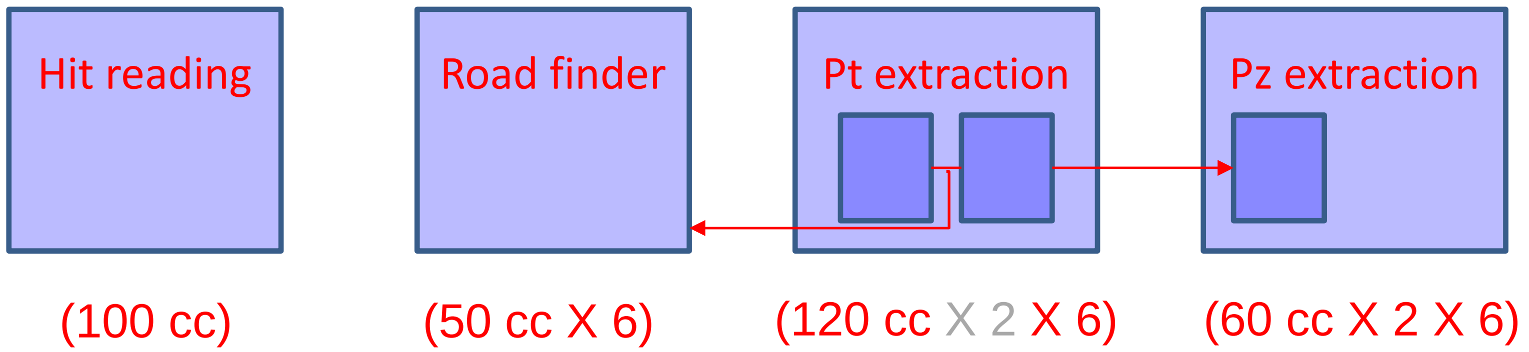
1) 1 GeV single muon
2) DPM

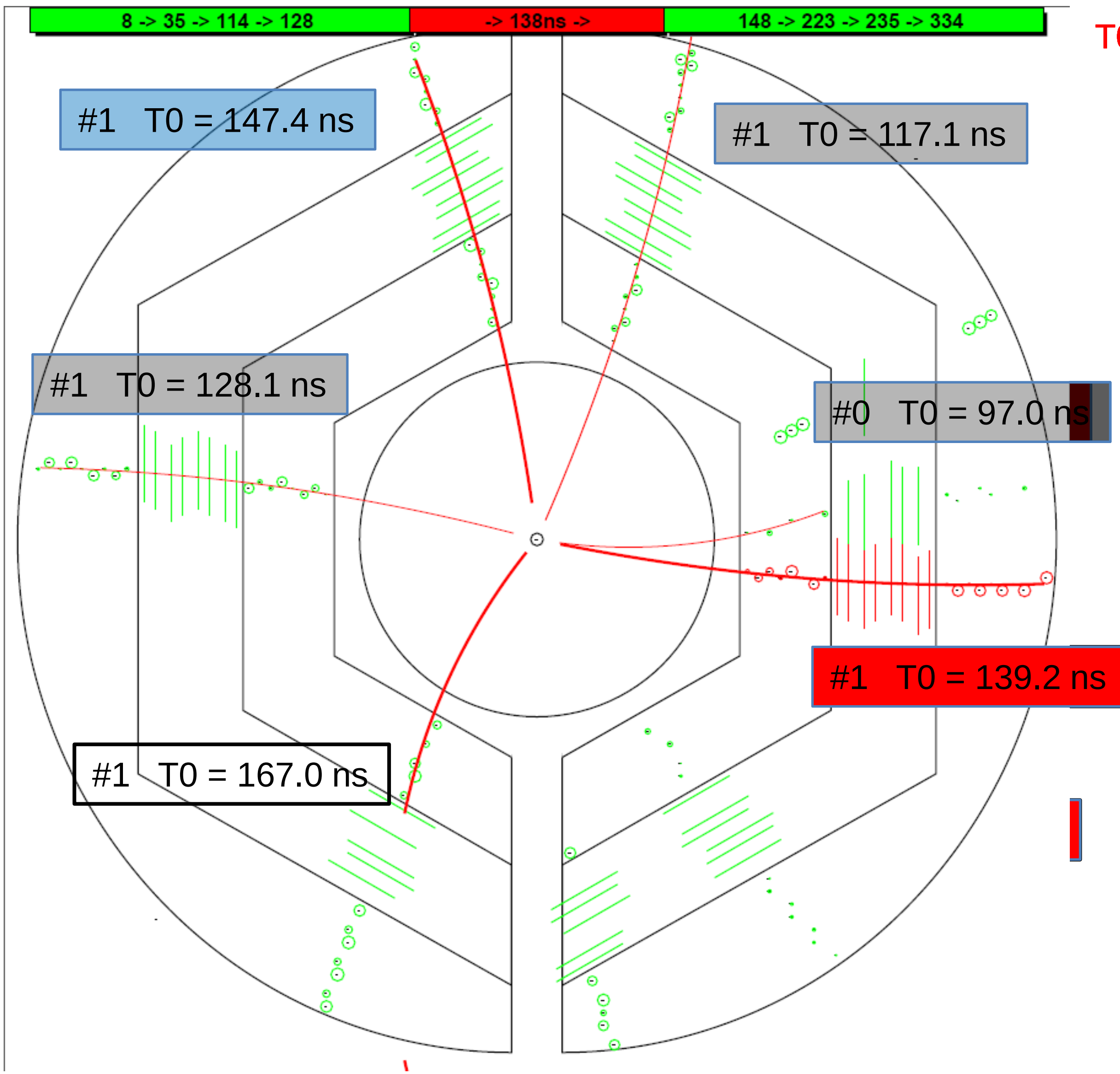
HK 22.7

Performance at FPGA

Device Utilization Summary			
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice Flip Flops	25,022	50,560	49%
DCM autocalibration logic	14	25,022	1%
Number of 4 input LUTs	33,120	50,560	65%
DCM autocalibration logic	8	33,120	1%
Number of occupied Slices	21,563	25,280	85%
Number of FIFO16/RAMB16s	148	232	63%
Number used as RAMB16s	148		
Number of DSP48s	124	128	96%

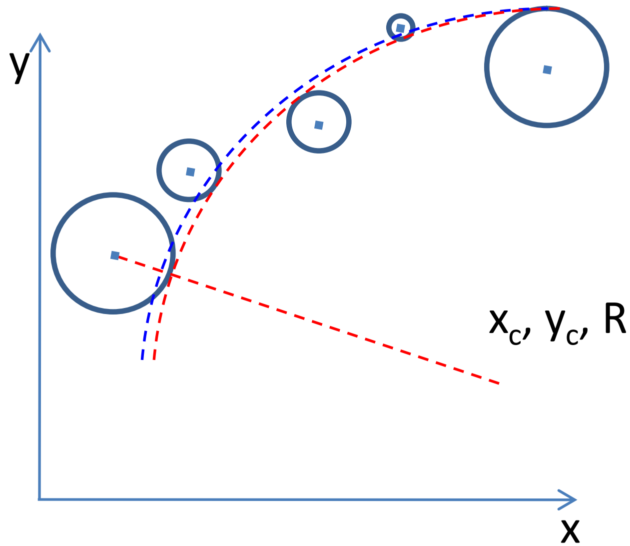
For one event with 100 hits (6 tracks): 7 μs



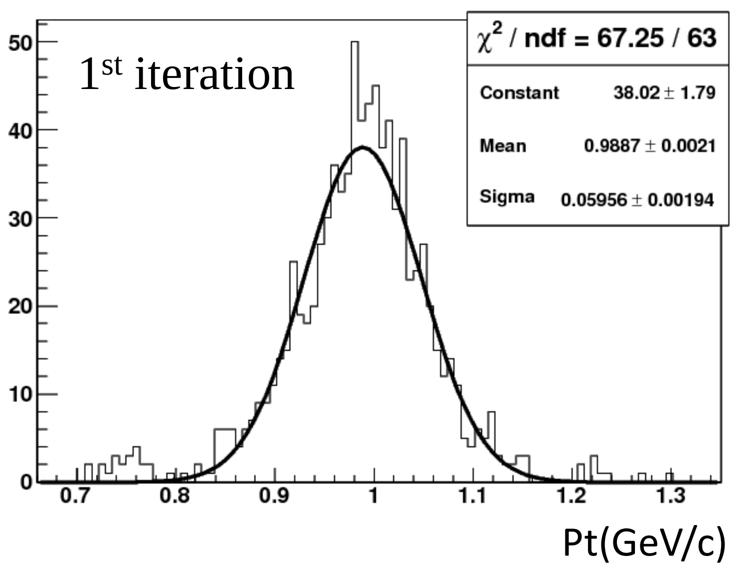


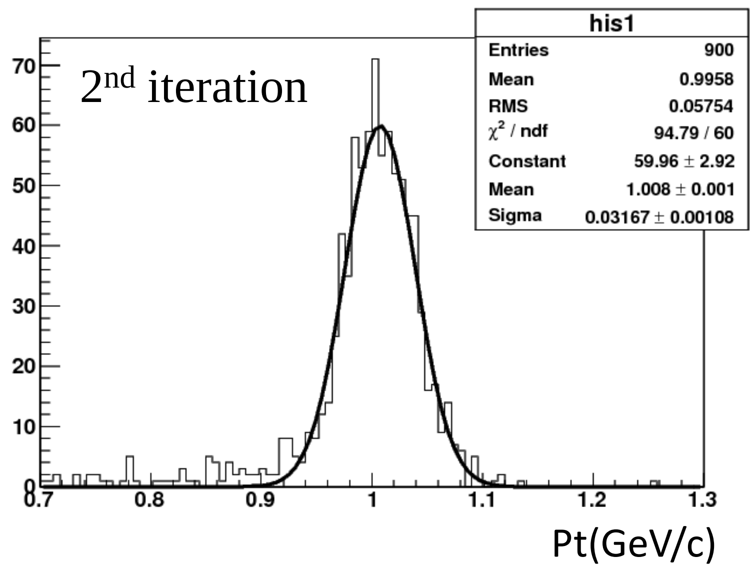
T0 (ns) : MC	Input	Iter_2
	0.0	
8.4	8.3	8.5
35.9	34.6	35.9
114.1	112.4	116.0
128.2	128.0	128.0
139.0	138.4	139.2
148.3	147.4	

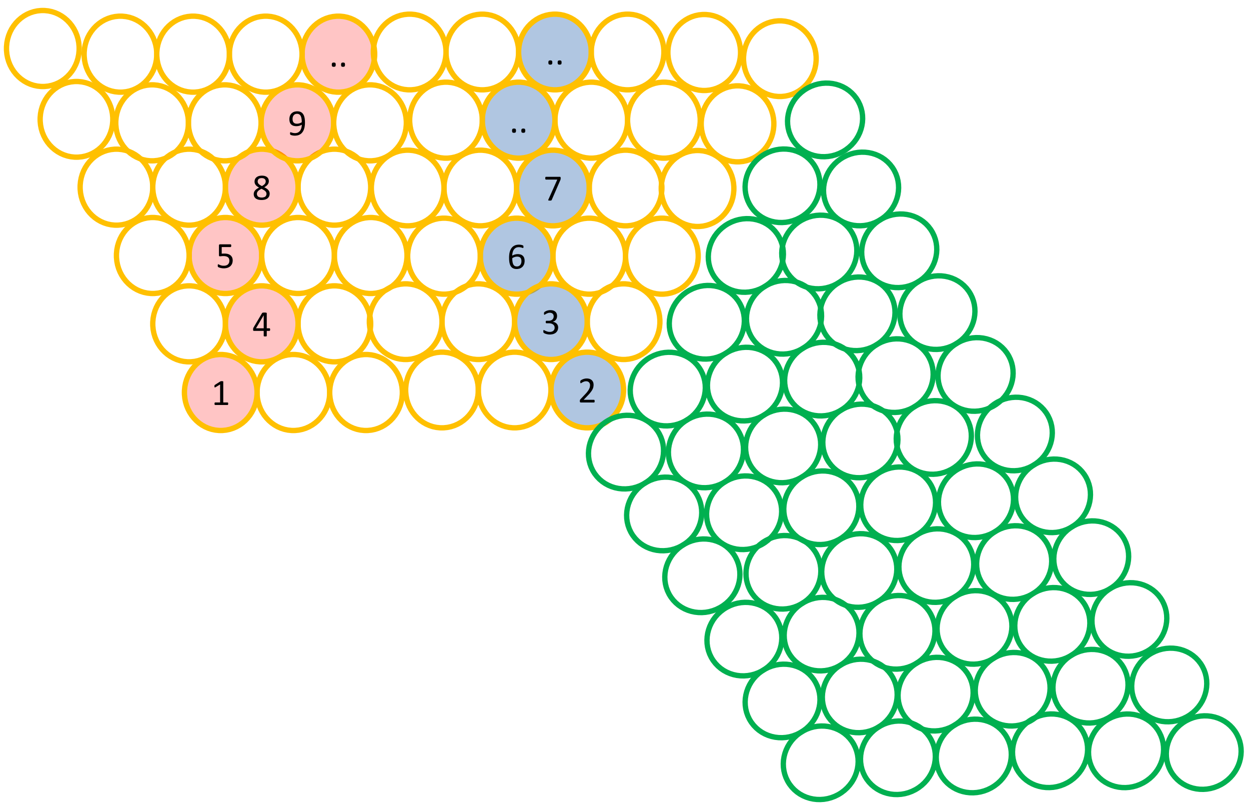
To Improve the Momentum Resolution -- using a 2nd iteration



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






..
..
9
..
8
7
5
6
4
3
1
2

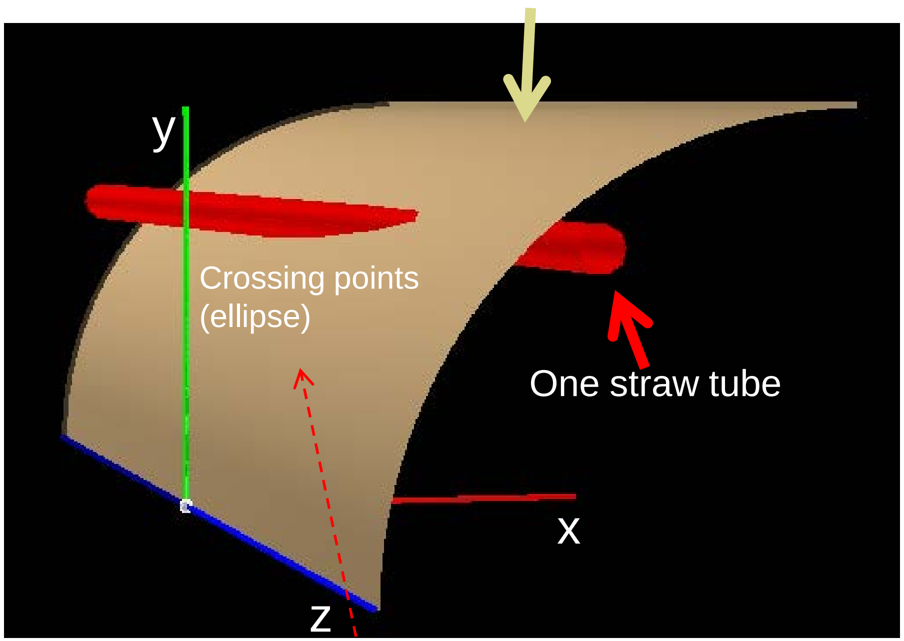
PZ reconstruction

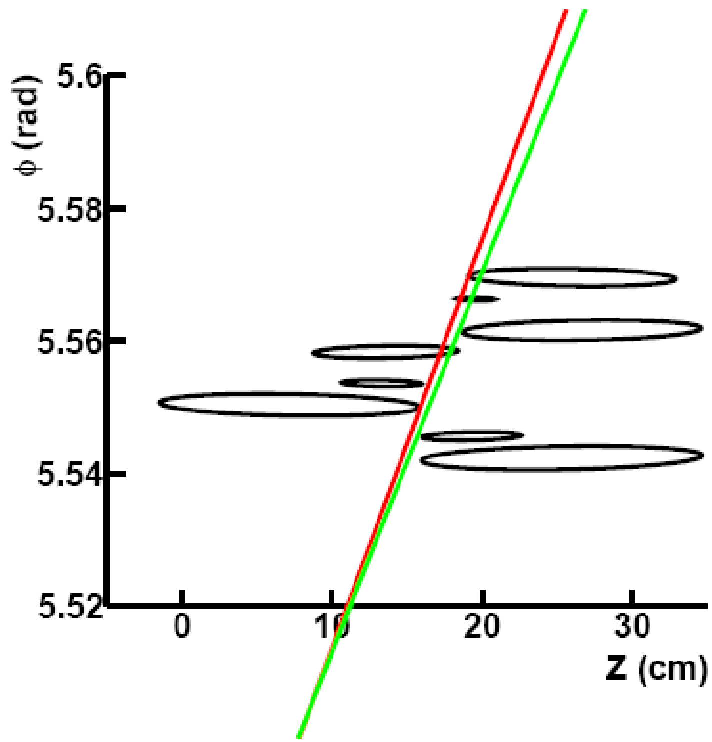
Pz reconstruction

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

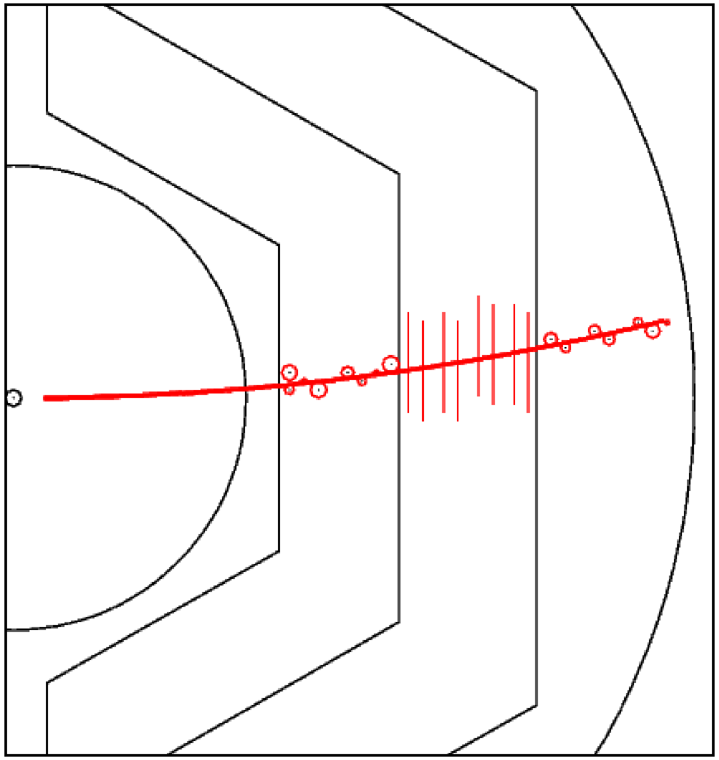


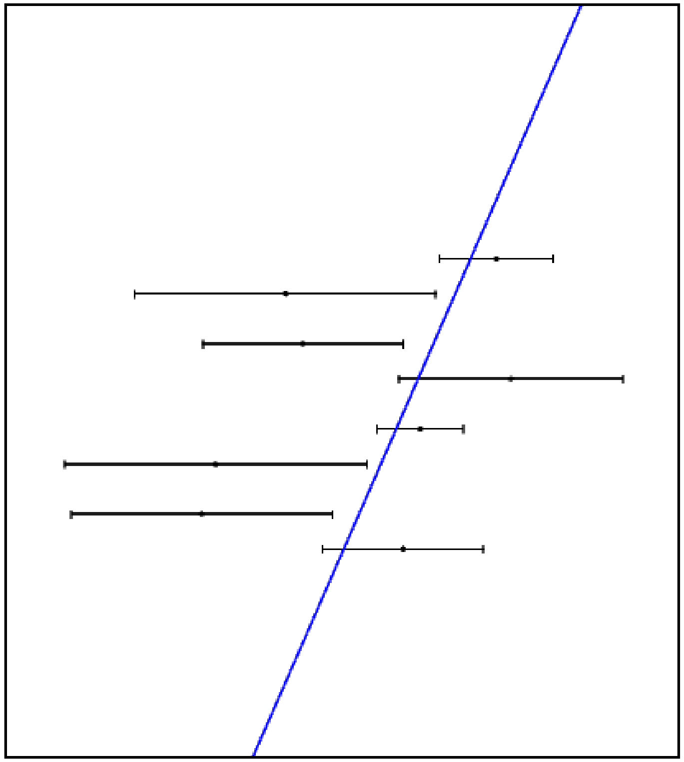


Track need to be tangent to the crossing ellipse.

Gianluigi Boca and Panda Collaboration, Panda STT TDR, 2012

Pz reconstruction





Known : z_i , Φ_i , d_i

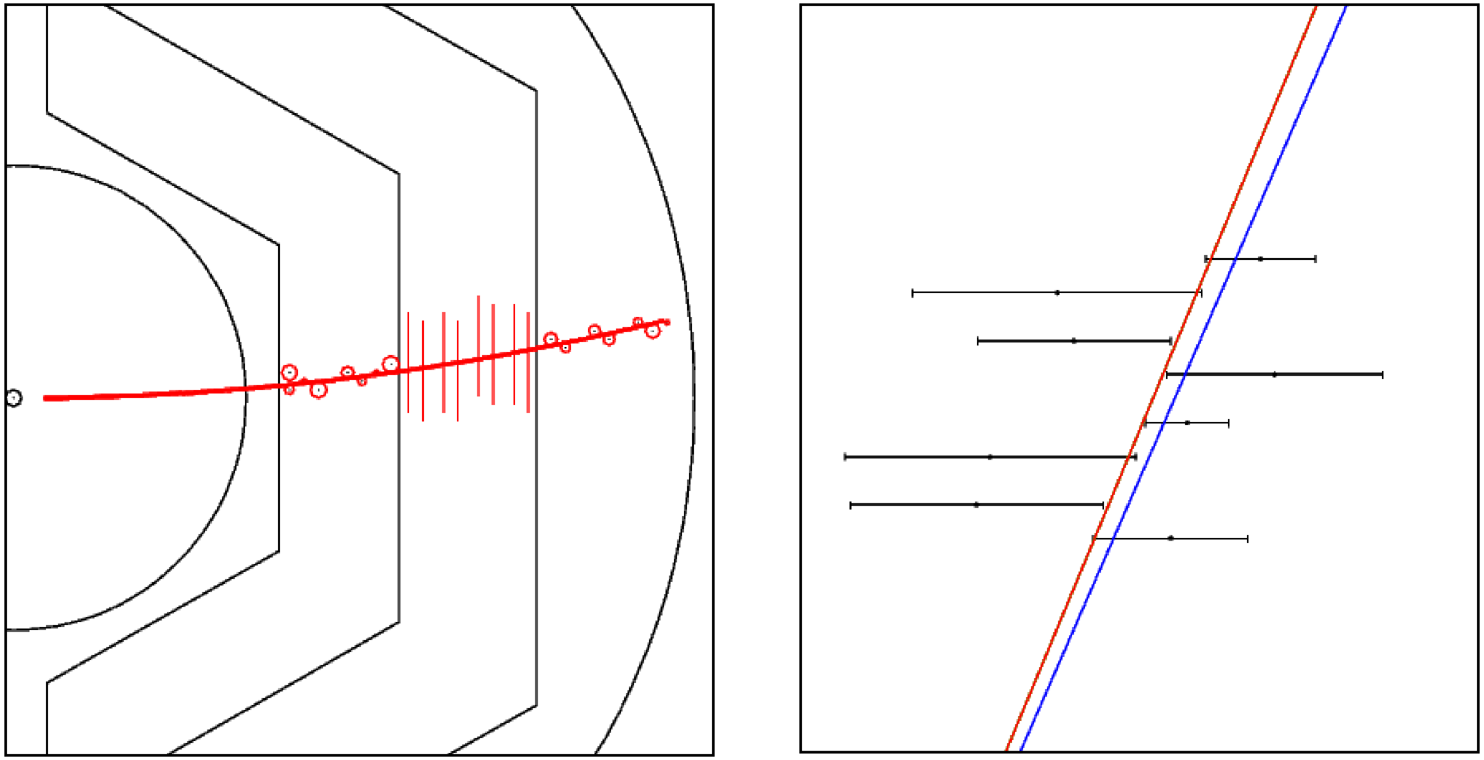
Question: To determine a line,

$\Phi + kz + \Phi_0 = 0$

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 %