Study of Retina Algorithm on FPGA for Fast Tracking

Wendi Deng and Zixuan Song
Université libre de Bruxelles(ULB)

21st IEEE Real Time Conference -Colonial Williamsburg

Introduction

Real-time track reconstruction in high energy physics experiments at colliders running at high luminosity is very challenging for trigger systems. To perform pattern-recognition and track fitting, artificial Retina or Hough transformation algorithms have been introduced in the field which have usually to be implemented in the state of the art FPGA devices.

We study two possible FPGA implementations of retina algorithm : one using online Floating-Point core and one using Look-up Table and fixed-point representation. Detailed measurements of the performance on hardware designs are investigated. So far the Retina has mainly be used in a detector configuration made of parallel planes, without or with weak magnetic field. Moreover we report on the simulated performance in a detector configuration made of concentric detection layers with high magnetic field (4T).

Retina Algorithm and Simulation

Retina algorithm is inspired from the processing of visual images by the brain where each neuron is sensitive to a small region of the retina. The strength of each neuron is proportional to how close the actual image projected on the retina region is to the particular shape that particular neuron is tuned to [1][2].

In a real HEP detector, the geometry of the detector and the topology of the events are quite complicated. To validate retina algorithm, we have used a simple tracker detector model made of 8 parallel tracking planes in the space without or with small magnetic field. We assume that every 3D trajectory of a charged particle is a straight line from the primary vertex (0,0,0) and identified by a pair of 2 parameters (x,y) in the plane. The (x,y) is the spatial coordinates of the intersection point of the track from the last layer (8th) . We discretize the last layer into a number of cells (patterns) 100* 100 , considering it as parameter space (u,v). The vertex (0,0,0) and the center of each cell (u_i,v_j) could identify an ideal track in the detector space uniquely, which means a set of straight lines with an array of the intersection coordinates over all layers are mapped in the space. The distance $\mathbf{s}_{i,j}^{k}$ of the intersections of the coordinates of the track $(\mathbf{x}_{i,j}^{k}, \mathbf{y}_{i,j}^{k})$ from the measured hits $(\mathbf{x}_k, \mathbf{y}_k)$ is computed (Figure 1). Then we are able to calculate the excitations **R** of each cell (u_i,v_j) following the function below :

$$R = \sum_k \exp\left(-\frac{{s_{i,j}^{k}}^2}{2\sigma^2}\right) \qquad (1)$$

where σ is a adjusted parameter for optimal response. The total response of the retina is obtained by calculating the excitations R of all cells. Finally tracks are identified by looking for a local maximum in the response array.

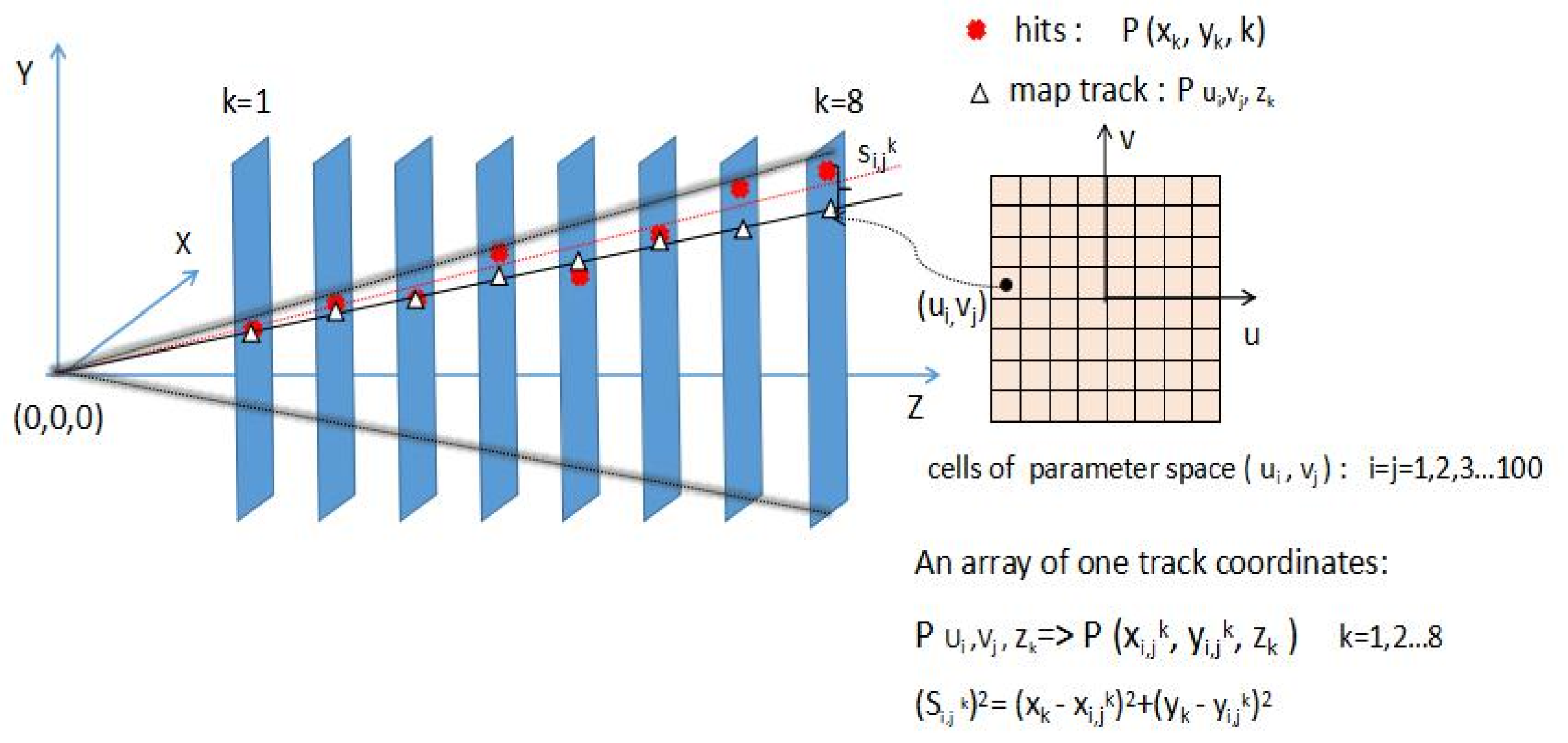

Y
X
Z
k=1
k=8
$s_{i,j}^{k}$
(0,0,0)
hits : $P(x_k, y_k, k)$
map track : $P_{u_i v_j z_k}$
v
u
(u_i, v_j)
cells of parameter space (u_i, v_j) : i=j=1,2,3...100
An array of one track coordinates:
$P_{u_i,v_j,z_k} \Rightarrow P(x_{i,j}^{k}, y_{i,j}^{k}, z_k)$ k=1,2...8
$(S_{i,j}^{k})^2 = (x_k - x_{i,j}^{k})^2 + (y_k - y_{i,j}^{k})^2$

Fig.1 3D detector model without M.F（magnetic field）

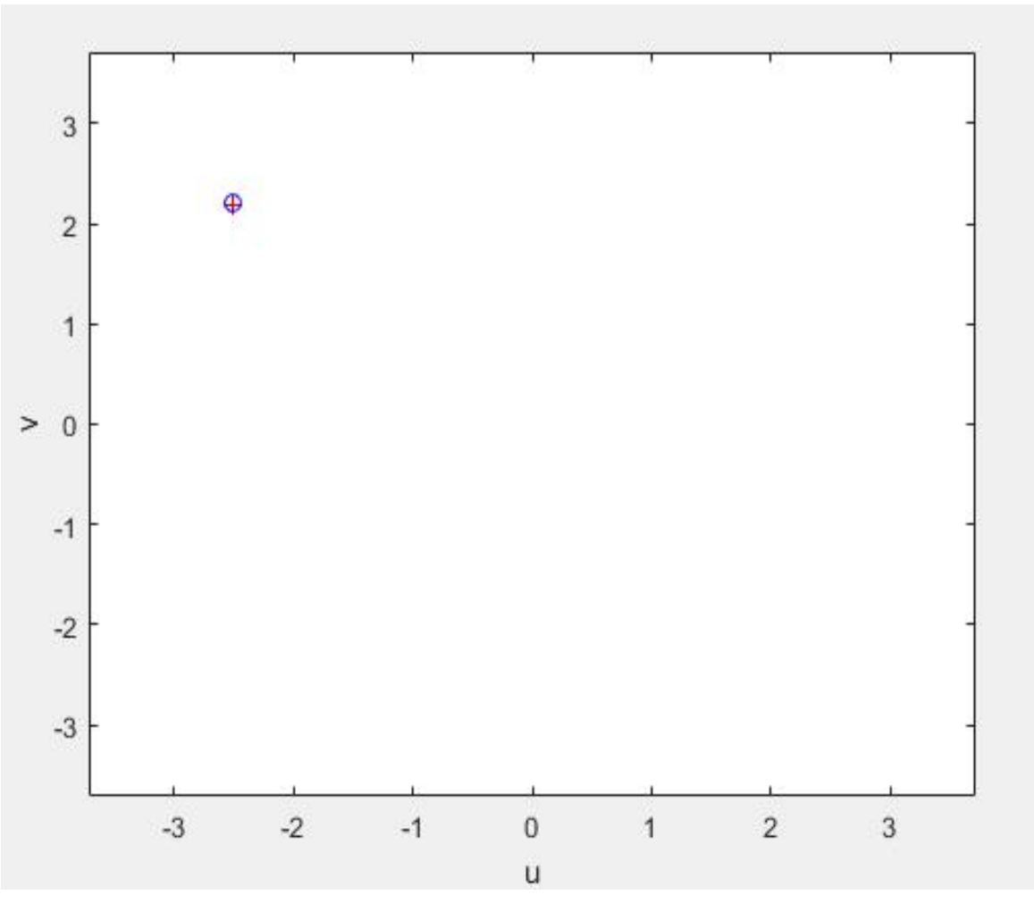


Fig.2 Comparison between reconstructed (blue circle) and generated (red cross) track parameters

Fig.2 describes the result of a simulation of above process. Generated parameter space (u ,v) consisting of 100 *100 space cells and random hits. A reconstructed track is identified and we could target the most possible (u_i,v_j) and compare with (x,y) by using retina algorithm.

RECO Tracks with magnetic field

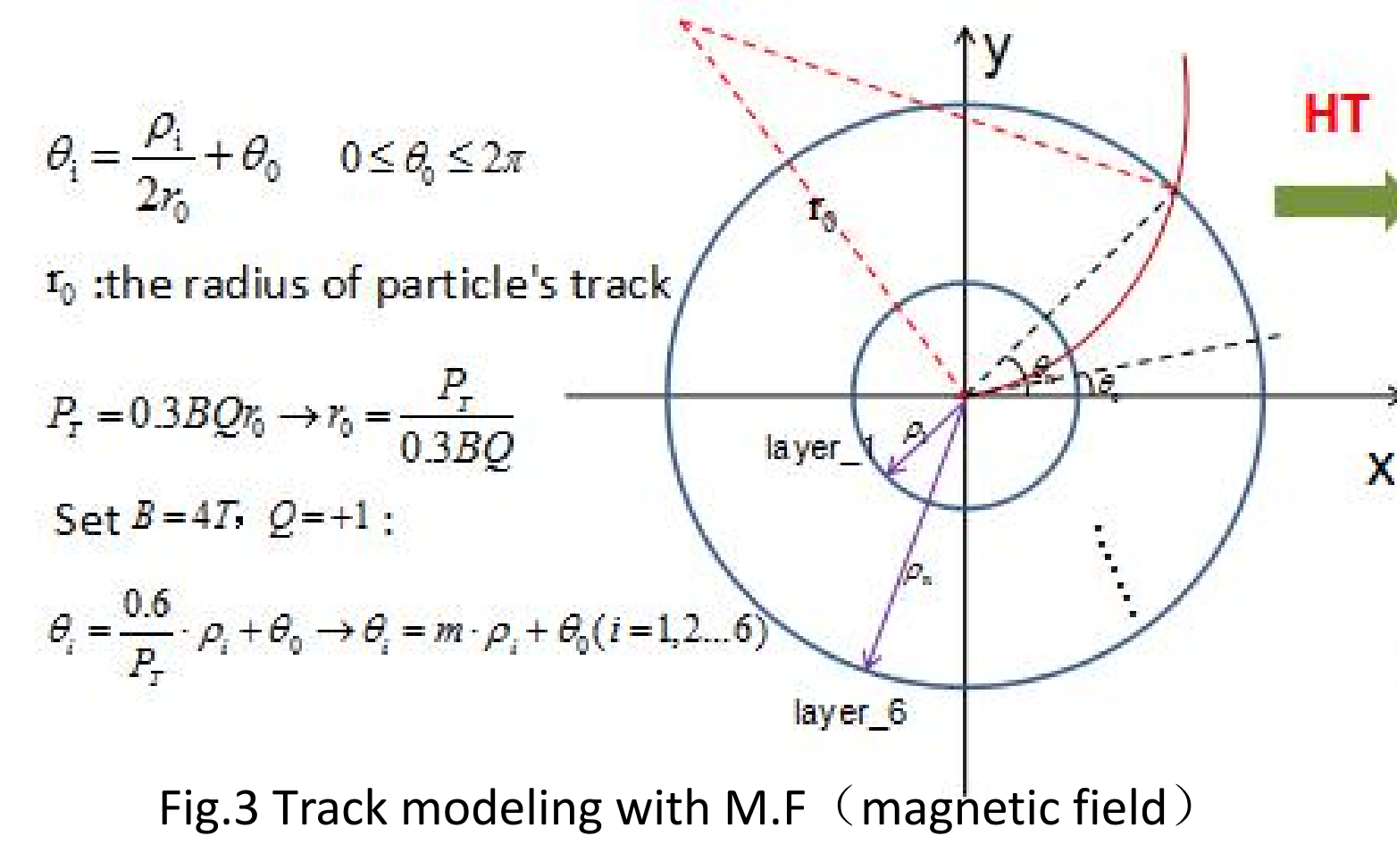

$\theta_i = \frac{\rho_i}{2r_0} + \theta_0 \quad 0 \le \theta_0 \le 2\pi$
r_0 :the radius of particle's track
$P_T = 0.3BQr_0 \to r_0 = \frac{P_T}{0.3BQ}$
Set $B=4T$, $Q=+1$:
$\theta_i = \frac{0.6}{P_T} \cdot \rho_i + \theta_0 \to \theta_i = m \cdot \rho_i + \theta_0 (i=1,2...6)$
y
x
r_0
layer_1
layer_6

Fig.3 Track modeling with M.F（magnetic field）

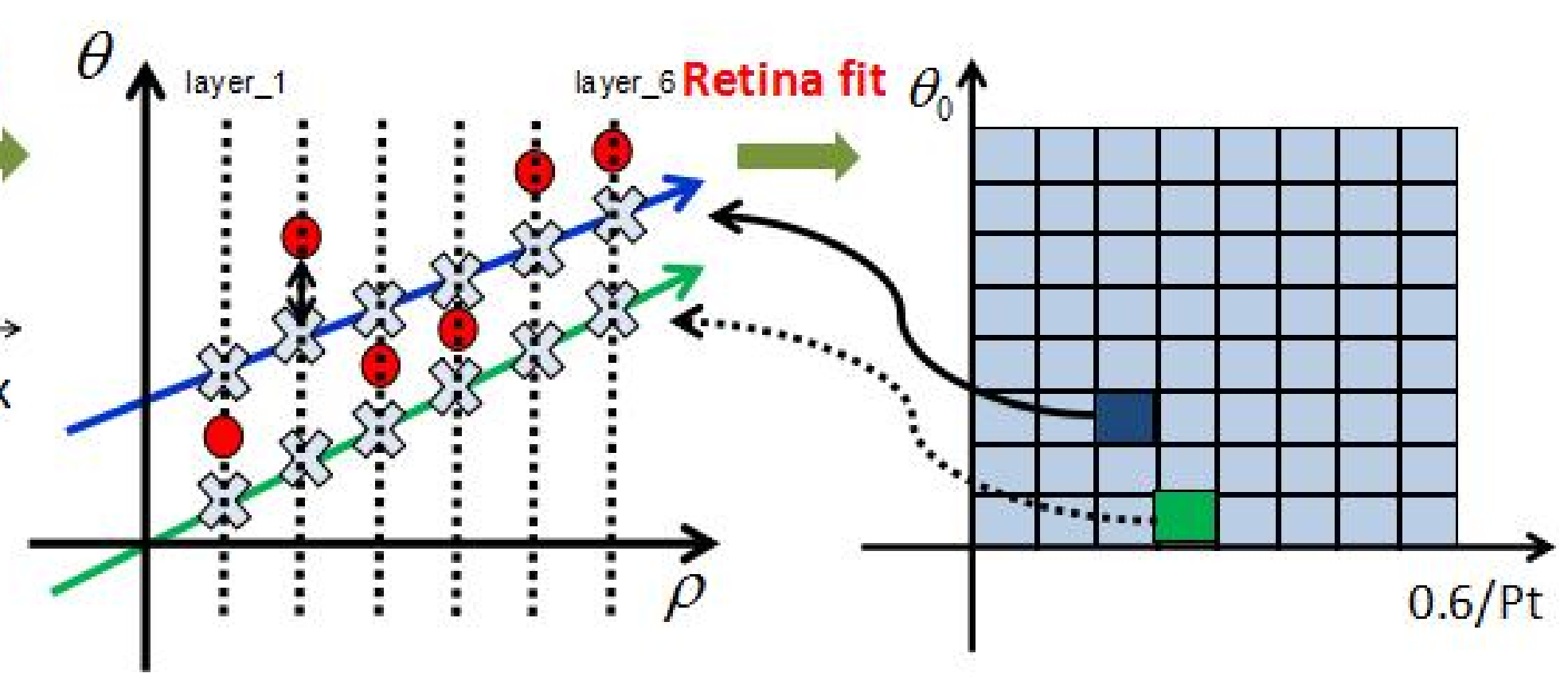
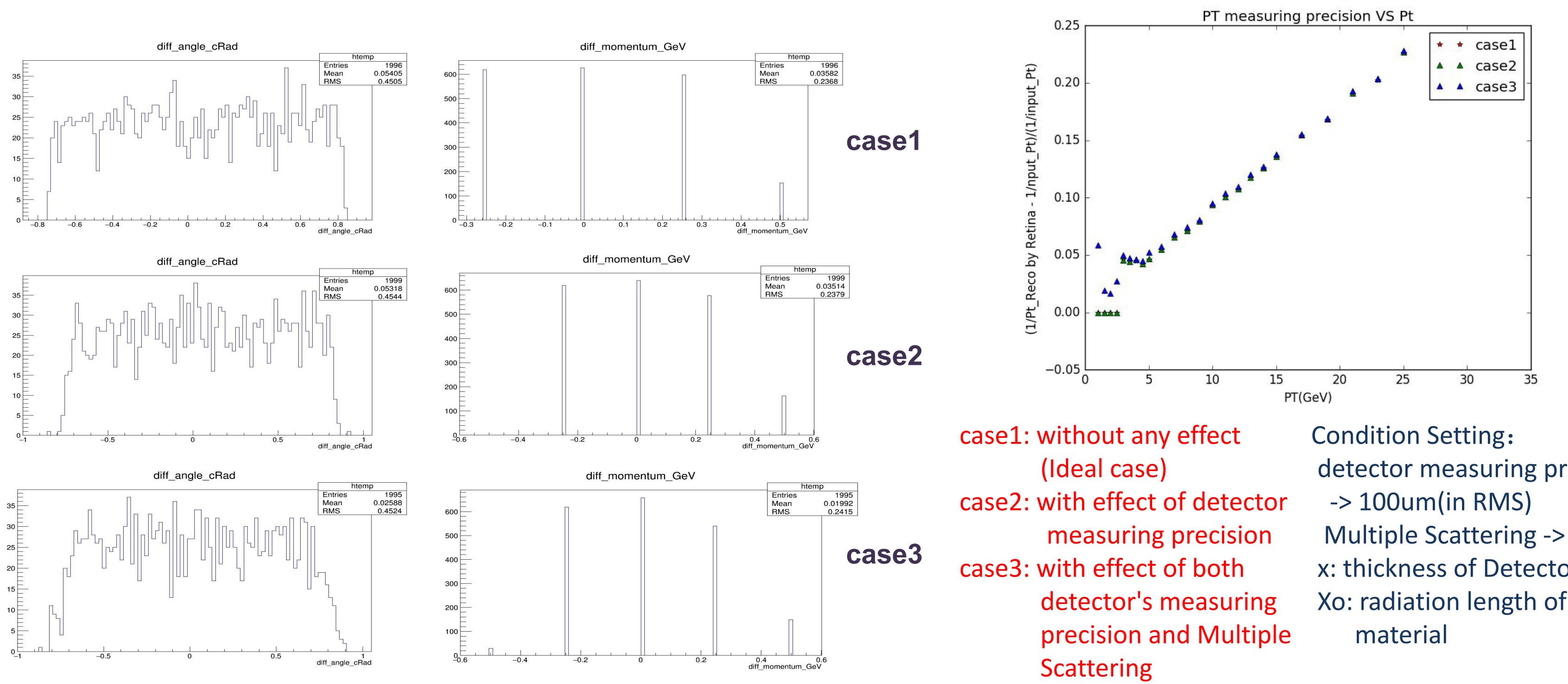


Fig.4 Parameter space (0.6/Pt, θ_0)

A more complex use case is the reconstruction of high Pt tracks in a barrel-like multi-layer detector in presence of a strong magnetic field. In the magnetic field the charge particle trajectory will bend (figure 3). The geometry is described as six (n=6) concentric circle layers with equal distance between them. The range of radius of those concentric circle layers is from 0.2 ($\rho_1 = 0.2$) meters (innermost) to 1.15 ($\rho_6 = 1.15$) meters (outermost). All the particles will start at the center of the circles with a given initial angle ,and go across six barrel layers from inner to outer. Due to the magnetic field effect, the shape of the track in this detector area is arc. Then using Hough Transformation change the parameter space into (0.6/Pt,θ_0) where Pt is the momentum of each charged particle and θ_0 is the initial direction angle of each track (figure 4) . In the end we use Retina to find optimal Pt and θ_0 for each individual track by scan the whole parameter space(cells) one by one. The Parameter range of Pt we set to scan by Retina is from 1GeV to 50GeV and for θ_0 is from 0 to 2PI (in radian). Whole parameter map is divided into 400*200 cells, while 400 degrees for Pt and 200 degrees for θ_0.



case1: without any effect (Ideal case)
case2: with effect of detector measuring precision
case3: with effect of both detector's measuring precision and Multiple Scattering

Condition Setting：
detector measuring precision -> 100um(in RMS)
Multiple Scattering -> x=0.1Xo
x: thickness of Detector material
Xo: radiation length of Det material

Fig.5 Graph of retina results resolution : (PT=5GeV,Number of bins 400*200,Pt measurement precision=(1/Pt_Reco by Retina - 1/input_Pt)/(1/input_Pt))

Fig. 5 shows the Pt reconstruction performance of the first run about the Retina simulations with magnetic field. We select fixed Pt of particles and random θ_0 $(0 \le \theta_0 \le 2\pi)$ as the input track event. Result in left shows the reconstruction Pt (output of Retina) distribution for each input track event with fixed momentum of particles , and the plot in right shows the Pt measurement precision against input_Pt with three different cases.

1. Zixuan.Song@ulb.ac.be 2. dengdandan@mails.ccnu.edu.cn

Hardware Design

To validate retina algorithm and measure its performance on FPGA, firmware prototypes have been designed and implemented. Our hardware design consists of 4 modules that mirrors retina processing: (1) Input data module: events and cells information load; (2) Distance module: distance $\mathbf{s}_{i,j}^{k}$ computation; (3) Exponent function module: weighing by exponent function to obtain response for each layer per cell; (4) ACC and comparator module: excitations of cells in the parameter space are accumulated and compared (shown in Fig.6).

In our previous work, our first implementation approach of retina is based on Floating-Point cores in the state of the art FPGA devices [3]. Our retina algorithm processor is made up of exponent function module and ACC and comparator module, which embed online Floating-Point Operator IP cores and a bus standard for on-chip communication AXI, providing rapidly and easily floating-point operators. With this first implementation we investigate latency and FPGA resource occupancy on KC705 platform (Kintex-7:7K325T-2FFG900). Finally our design based on Floating-Point cores fills at a maximum 70% of the FPGA resource up to six cells processing parallely and takes a latency of 197 [3]. FPGA performance keeps improving the number of gates and the interconnection speed, therefore the devices could definitely improve the performance making the Floating-Point Operator core attractive in the future as they are very flexible and can offer higher precision. Never the less, we have to optimize our design of Floating-Point cores in a more conventional way and improve ACC and comparator module with fixed-point calculation to decrease the latency and FPGA resource cost. One step further, as most commercial FPGA designs are limited to finite precision signal processing using fixed-point representation, we decided to optimize retina on FPGA more efficiently and economically with a fast solution by full fixed-point calculation. We implement input data module and distance module and use Look-up Table (LUT) approach instead of Floating-Point cores within exponent function module , in which function values are pre-calculated at certain sample points and stored in memory. At this current stage, we follow the same hardware architecture of six cells when developing fixed-point & LUT design and improve Floating-Point core design for further comparison.

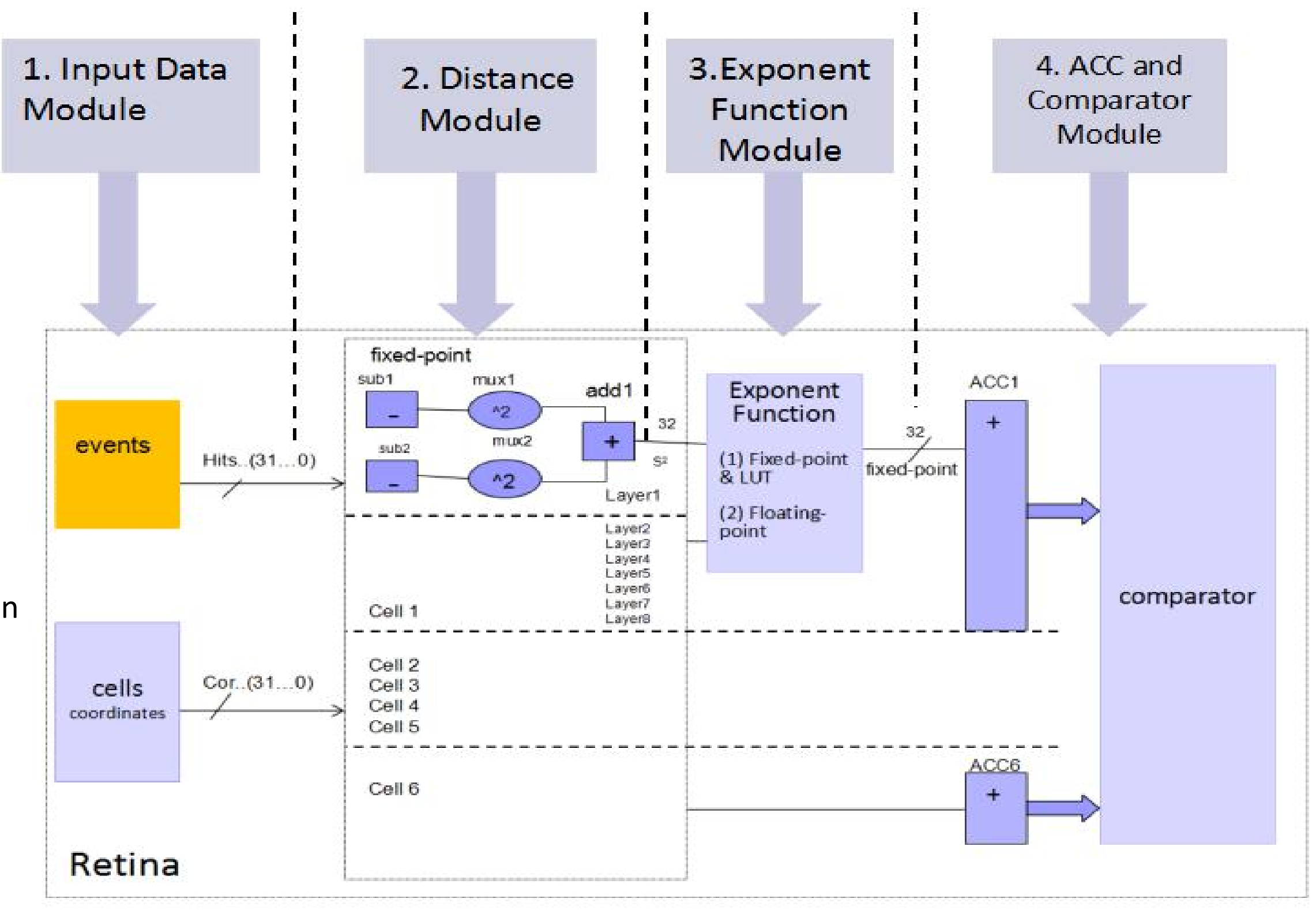


Fig.6 Firmware design architecture of retina algorithm

Table.1 FPGA resource usage and latency for two firmware designs

Clock (Hz)	Firmware Design	DSP (%)	LUT (%)	LUTRAM (%)	BRAM (%)	FF (%)	Latency (Cycles) /μs
100M	Floating-point	17.14	70.72	15.24	3.03	42.32	156/1.56
100M	Fixed-point and LUT	11.43	7.25	8.46	33.48	9.8	68/0.68

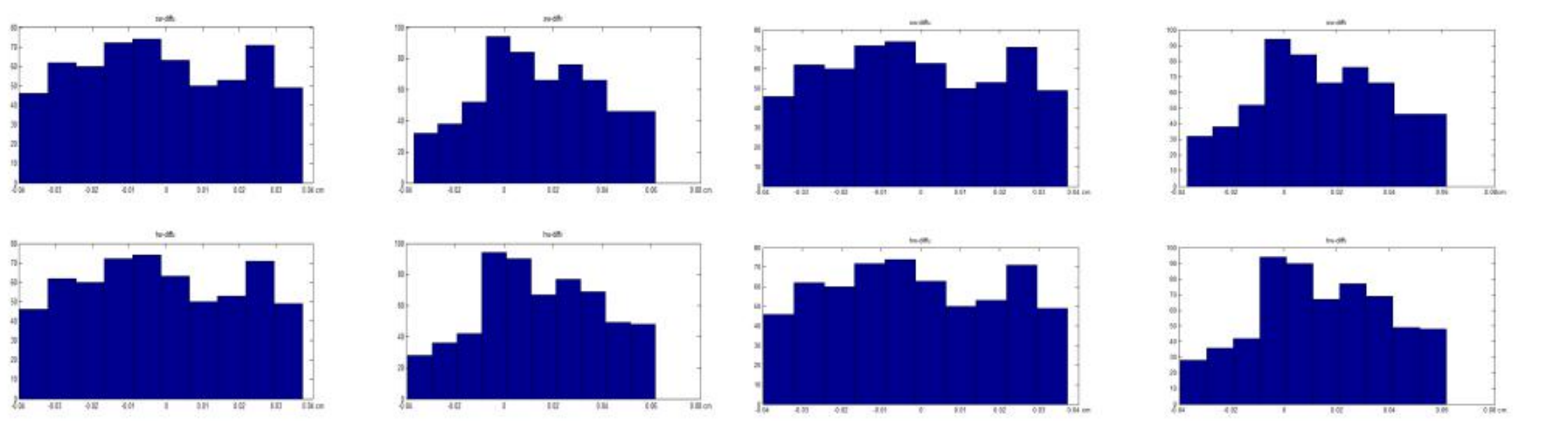
In Table.1, we compare the performance for two firmware designs, under the same test conditions, our fixed-point and LUT firmware typically reduce the latency by a factor 2 and resource usage.

Performance

events: 600
hits granularity: 0.0495cm; parameter space (u,v) granularity: 0.0742 cm

Fixed-point & LUT firmware resolution:
Q12 1/(2^12)=0.00024

Floating-point firmware resolution:
1) Fixed-point Q12: 1/(2^12)=0.00024
2) Floating-point: exponent: 8 bit; fraction: 23 bit; Precision: 0.000001



sw: software
hw1: fixed-point & LUT
hw2: floating-point

	sw_diffu	hw1_diffu	hw2_diffu	sw_diffv	hw1_diffv	hw2_diffv
RMS	0.0213	0.0213	0.0213	0.0283	0.0283	0.0283
mean	-0.0013	-0.0013	-0.0013	0.0139	0.0138	0.0138

Fig.7 Comparison of the spatial resolution (cm) in the (u,v) plane between both hardware implementations against the software simulation. Both hardware implementations use a 12-bit fixed-point representation. The floating-point processing unit use 8-bit exponent and 23-bit fraction

Q12	sw_diffu	hw_diffu	sw_diffv	hw_diffv
RMS	0.0213	0.0213	0.0283	0.0283
mean	-0.0013	-0.0013	0.0139	0.0138
Q10	sw_diffu	hw_diffu	sw_diffv	hw_diffv
RMS	0.0220	0.0220	0.0289	0.0289
mean	-0.0027	-0.0027	0.0130	0.0130

Fig.8 Comparison of the spatial resolution (cm) in the (u,v) plane between 12-bit fixed-point and 10-bit fixed-point representation for the fixed-point & LUT firmware design

In Fig.8, we use two resolution of fixed-point data computation in fixed-point & LUT design. The shown results illustrate that from Q12 to Q10 resolution declines, as well as RMS of diffu/diifv, which verifies computation of the design is right. In the fixed-point & LUT design, we keep the resolution Q12 for fixed-point.

To quantify retina resolution , we compare these two approaches results with software simulation. (diffu/diffv means the differences between the (u,v) pair of which hits are generated and the reconstruct (u,v) in the parameter plane mapped by retina). These outcomes indicate that both approaches can find out the cell candidate. Considering the FPGA resource cost and latency, fixed-point & LUT based design offers a better choice.

Conclusion and Outlook

In this note, we study FPGA-based implementations of the artificial retina algorithm for fast track reconstruction in trigger system. So far, we apply retina to a simple detector ignoring magnetic field effects and present on KC705 using both Floating-Point IP and fixed-point &LUT approaches. The performance of implementations including latency, resource, algorithm precision performance have been compared as well, which can be estimated to a complete prototype of hardware system scale.

Moreover, our research is targeted to adapt retina algorithm to a more realistic tracker detector with cylindrical geometry. Due to the magnetic field effects, charged particle trajectories in this detector are bent and treated as partial arc. A first retina modelling of track reconstruction under the situation of particles in a magnetic field with six barrel layers tracker has been built. Our purpose is to find out the optimal configuration parameter to balance the size of parameter space and measurement resolution of Retina. Then we will mirror this modelling on FPGA to evaluate hardware performance and whether Retina is suitable for fast tracking under magnetic field in realistic CMS experiment.

References

[1] L.Ristori, “An artificial retina for fast track finding,” Nucl. Instrum. Meth. A 453 (2000) 425.
[2] A.abba et al., “The artificial retina processor for track reconstruction at the LHC crossing rate,” JINST 10, C03018 (2014), [arXiv: 1409.1565].
[3] Z.Song et al., Study of hardware implementation of fast tracking algorithms, 2017 JINST 12 C02068.