UNIVERSITÉ LIBRE DE BRUXELLES

Study of Hardware Implementation of Fast Tracking Algorithms

CENTRAL CHINA NORMAL UNIVERSITY

Zixuan SONG*, Gilles DE LENTDECKER, Yifan YANG, Frédéric ROBERT,
Guangming HUANG, Alexandre LÉONARD, Jianmeng DONG

*email : Zixuan.Song@ulb.ac.be

Introduction

Real-time track reconstruction at high event rates is a major challenge for future experiments in high energy physics. To perform pattern-recognition and/or track fitting, artificial retina or Hough transformation have been introduced in the field which have to be implemented in the FPGA firmware.

We study on a possible FPGA hardware implementation of retina algorithm based on the Floating-Point Operator IP. Detailed measurements with the algorithm are investigated. Retina performances and capabilities of the FPGA are discussed along with perspectives for further optimizations and applications.

Retina Algorithm

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.

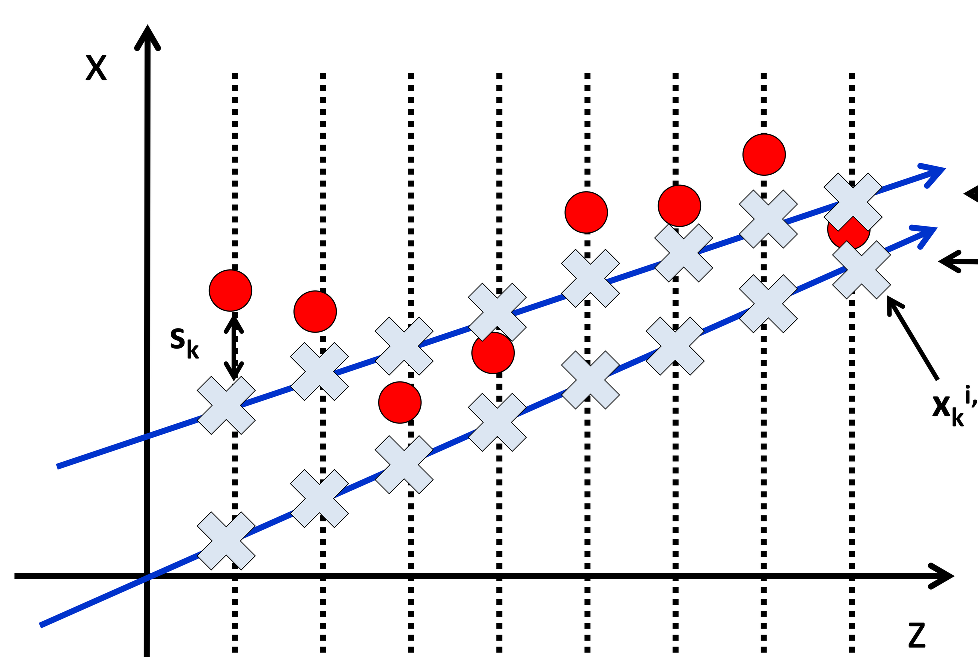


Fig.1.1 tracking plane (z,x)

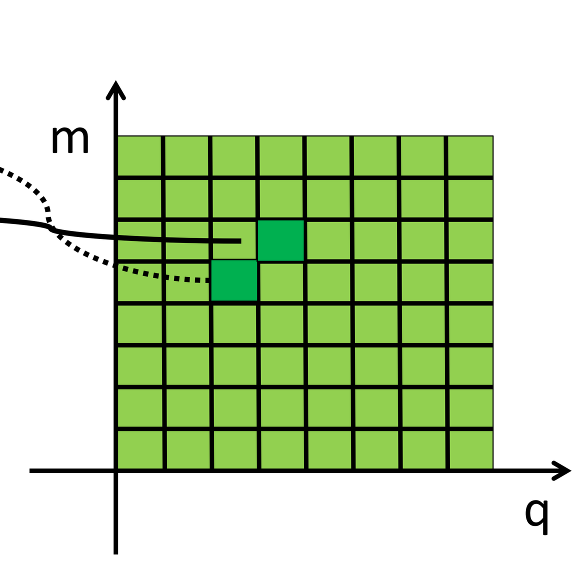


Fig.1.2 parameter space (m,q)

To apply this concept in particle physics, let's take a simple experiment made of 8 parallel layers of a position-sensitive detector providing 1D particle coordinates **x**, and where the particle tracks are straight lines (x(z) = m * z + q). We discretize the parameter space (m,q) into cells (figure1.2). The center of each cell identifies an ideal track in the detector space uniquely (figure1.1). Therefore the parameter space cell (m_i, q_j) maps into a series of detector cells $\mathbf{x}_k^{i,j}$, where k is the layer. For each event, the distance $\mathbf{s}_k$ of the intersections of the coordinates of the track $\mathbf{x}_k^{i,j}$ from the measured hits is computed and the response of the space cell (m_i, q_j) is calculated.

$$R = \sum_k \exp\left(-\frac{s_k^2}{2\sigma^2}\right) \qquad (1)$$

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

Retina processing flow

1. Distance : compute distance s between hits and track coordinates
2. Weight : calculate the weights associated to these distances (set σ = 1 as a constant)

$$w = \exp\left(-\frac{s_k^2}{2\sigma^2}\right) \qquad (2)$$

3. Accumulator : accumulate these 8 weights
4. Maximum : check if it is a local maximum
5. Output : send out the max value and corresponding space cell

Retina Simulation

Fig.2 describes the result of a simulation of this process. Generated parameter space (m_i ,q_j) consists of 201*201 space cells and random hits; we process retina with MATLAB. A graphical rendering of the responses is shown: each small grid correspond to one cell and the intensity of the responses as calculated in Eq.(1) is represented by different levels of colours. A reconstructed track is identified by a local maximum.

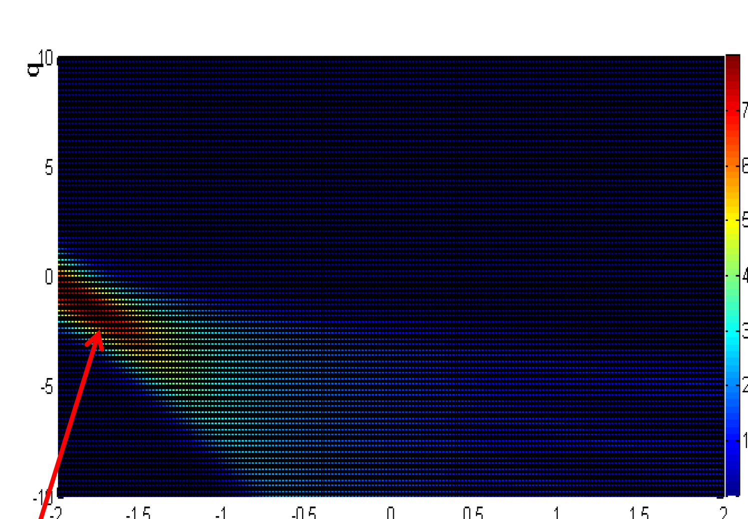
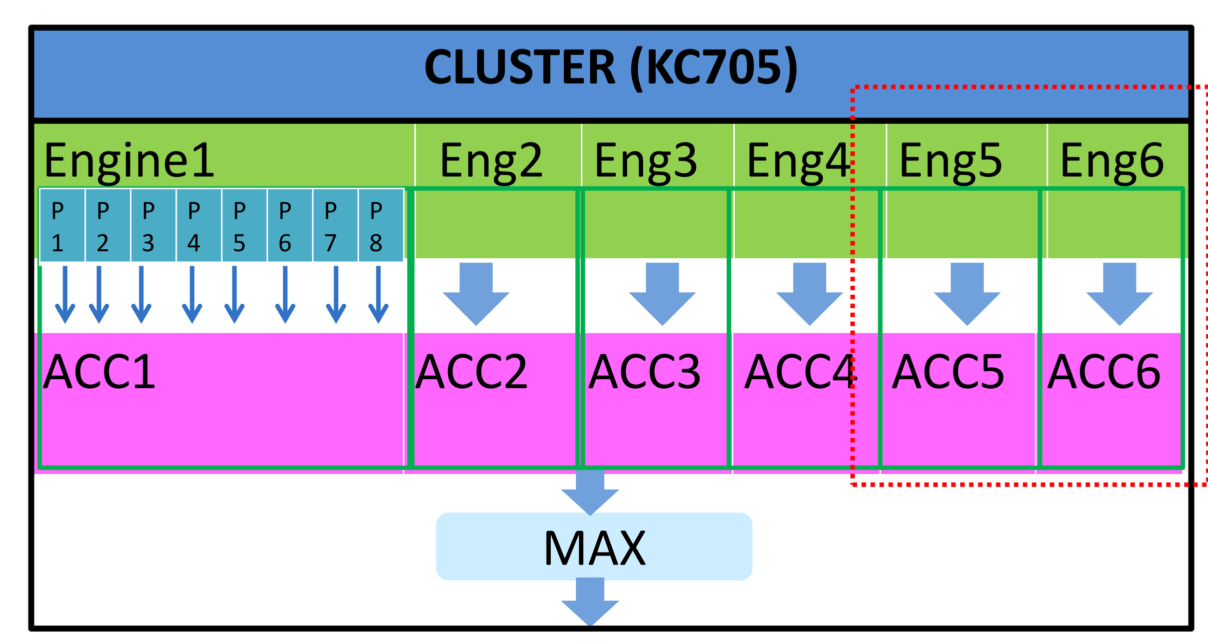


Fig.2 retina simulation

Hardware Implementation

Our approach follows the current trends of industry using high-speed, high-bandwidth commercially available FPGA devices. The architecture of our design mirrors the retina processing description (figure 3). We fix encoding with a group of 32 bits floating-point distances as input constant in the firmware. The weights associated to the distances with 8 planes is defined as a logic module, named processing. Each processing is implemented as an independent block that performs the necessary arithmetic operations with Floating-Point Operator IP. Data flow is delivered directly to all processing. The array of space cells is mapped into a series of processors, the engine. Each engine evaluates and accumulates the excitation of 8 processings in parallel. Therefore local maximum is found in the engines. Track parameters are extracted from the cell space finally.



P1-P8 : processing
Eng1-Eng6 : engine
ACC1-ACC6 : accumulator
MAX : maximum

Fig.3 design architecture

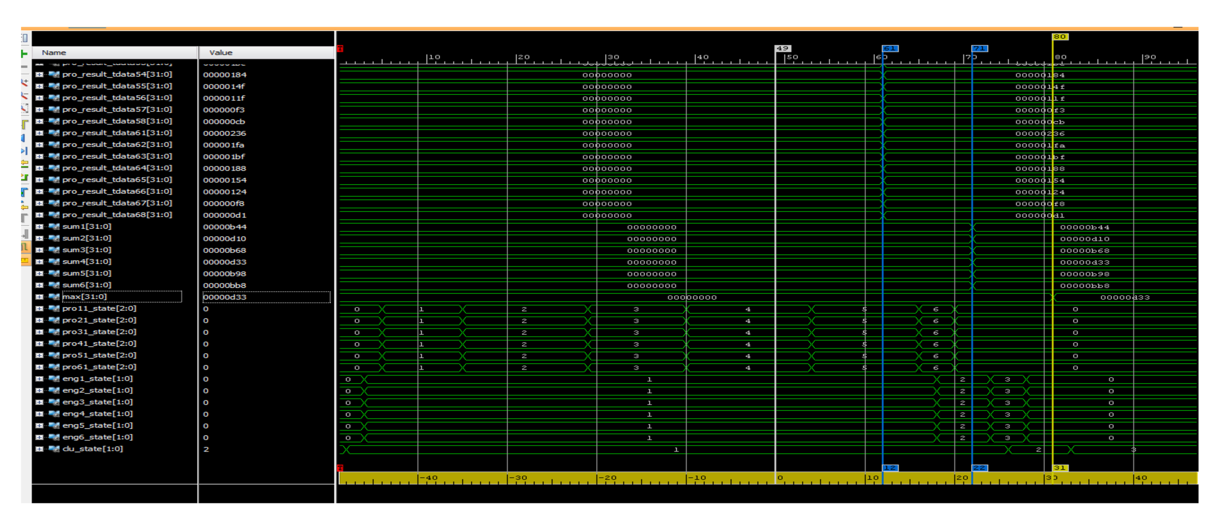


Fig.4 observerd results through ILA (chipscope)

A KC705 platform can process maximum 4 engines (cells) with Floating –Point data per period in parallel. To optimize latency and serve more engines effectively, we transform Floating - Point input into Fixed -Point data after processing blocks. Figure.4 shows the latency information for Fixed –Point calculation.

Performance

Retina was tested on different hardware platforms including Artix-7 (7A200T-2FFG1156) and Kintex-7(7K325T-2FFG900). The full simulation indicated that the devices sustain an input frequency of 100MHz clock, with FPGA resource occupation and contributions to latency as displayed in Table.1.

Table.1 FPGA occupation and latency

Clock (Hz)	Eng	DSP (%)	LUT (%)	LUTRAM (%)	FF (%)	Processing	Fix	ACC	Max	Latency(Cycles) /µs
100M(K7)	4	13.81	49.4	9.37	14.63	49	0	114	25	188/1.88
100M(A7)	4	21.56	55.93	21.87	31.96	51	0	117	27	195/1.95
100M(K7)	6	20.71	72.24	13.95	21.71	49	0	114	34	197/1.97
100M(K7)	6	17.14	69.86	8.14	20.00	49	12	10	8	80/0.8

Blue regions demonstrate different placing capacity between A7 and K7. In Floating-Point and Fix-Point mode, contributions to latency are listed in red section. We leave approximately 30% of logic available for other uses. Pipelined architecture with dynamic data flow will be achieved in the next step.

Conclusion and Outlook

Our research focuses on the study of the possible implementation and hardware optimization of the retina algorithm for fast track reconstruction in particle physics. We have successfully verified that retina in real time using highly parallelized computation is suitable for implementation in FPGA. With entire triggering chain latency and events buffer limitation, the algorithm is very flexible in terms of calculation process and latency so that better performance can be improved in our design.

From above, these results are encouraging and represent a first step towards the development of fast efficient track-fitting system for applications in future experiments at high luminosity LHC. However, there is a significant and complex leap to pass from a proof of concept in a bidimensional ideal case to application in an experiment like CMS with a cylindrical geometry at HL-LHC.

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].