A specialized processor for track reconstruction at the LHC crossing rate

A. Abba[2], M.Citterio[2], F.Caponio[2], A. Cusimano[2], A. Geraci[2]
P. Marino[3], M. Morello[3], N. Neri[2], A. Piucci[3], M. Petruzzo[2], Giovanni Punzi[3],
L.Ristori[3,4], F. Spinella[3], S. Stracka[3], D. Tonelli[1]

[1]CERN [2]Politecnico/INFN-Milano [3]University/INFN-Pisa [4]Fermilab

INSTR 2014
BINP, Novosibirsk, Russia

Motivation

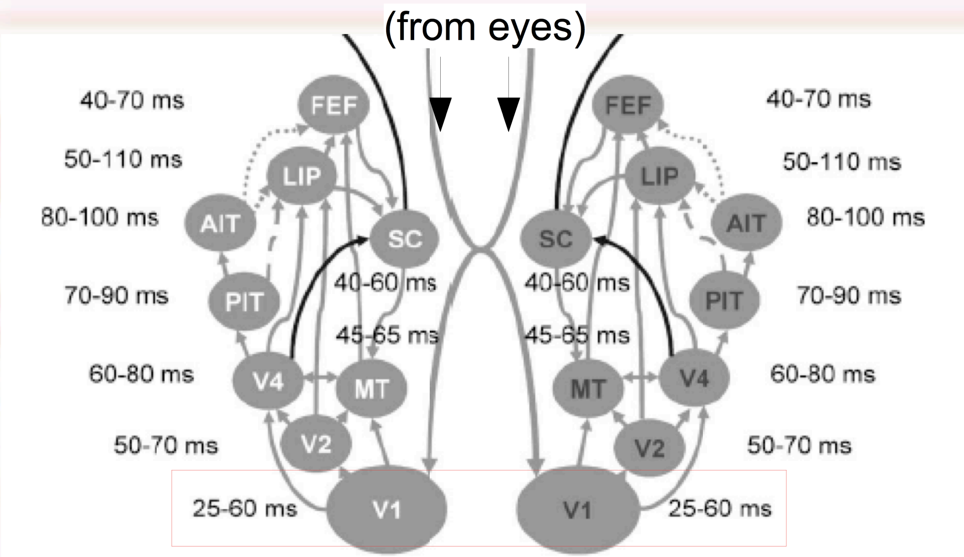
- The LHC has opened a new era, also about instrumentation
- Exploitation of HL will pose even greater challenges
- Data acquisition and reconstruction one of the toughest issues
- A big part of the problem is the reconstruction of charged particle trajectories
  - Large combinatorial problem, calls for high parallelization
  - In many cases, latencies are an issue due to need for buffering (e.g. in CMS tracker).

Some past examples of real-time track reconstruction

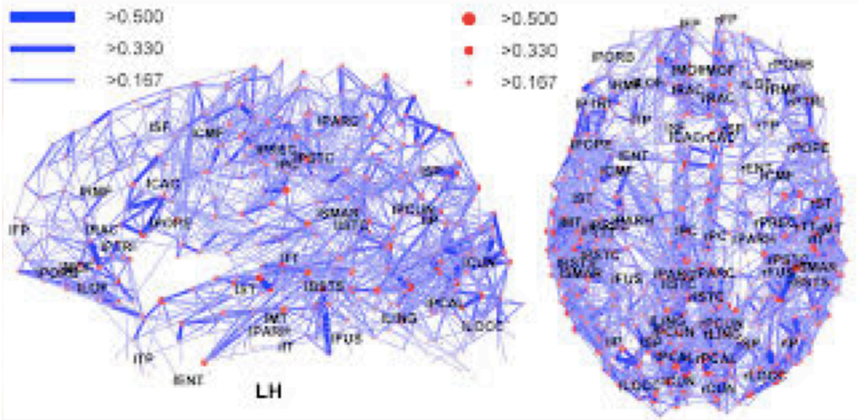
Name	Tech.	Exp.	Year	Event rate	clock	cycles/event	latency
XFT	FPGA	CDF-L0	2000	2.5 MHz	200 MHz	80	4μs
SVT	AM	CDF-L2	2000	0.03 MHz	40 MHz	~1600	<20μs
FTK	AM	ATLAS-L2	2014	0.1 MHz	~200 MHz	~2000	O(10μs)
Compare with the requirements of a L0@LHC:							
?	***?***	***LHC-L0***	***~2018***	***40MHz***	***~1GHz***	***~25***	***few μs***

- The task of L0 tracking at LHC appears daunting despite the progress of electronics.
- Any complex tracking calls for $O(10^3)$ clock cycles/event (both in latency and throughput)
- No known example of a system making a non-trivial pattern reconstruction in O(25) time units

Well, maybe I can think of ONE example...



Adapted from H. Kirchner, S.J. Thorpe / Vision Research 46 (2006) 1762–1776



- The early visual areas (V1) in human brain produce a recogniz-able sketch of the image in ~30ms
- The maximum neuron firing frequency is ~1kHz → **~30 t.u**
- Far-fetched example ? See [Del Viva MM, Punzi G, Benedetti D PloS one (2013) - DOI: 10.1371/journal.pone.0069154] experimental evidence that V1 functionality can be quantitatively modeled as a “trigger”.

What's special about the “brain algorithm” ?

- Parallelism, of course - but SVT and FTK are based on Associative Memories, that are very parallel devices as well...
- Two important differences, though:

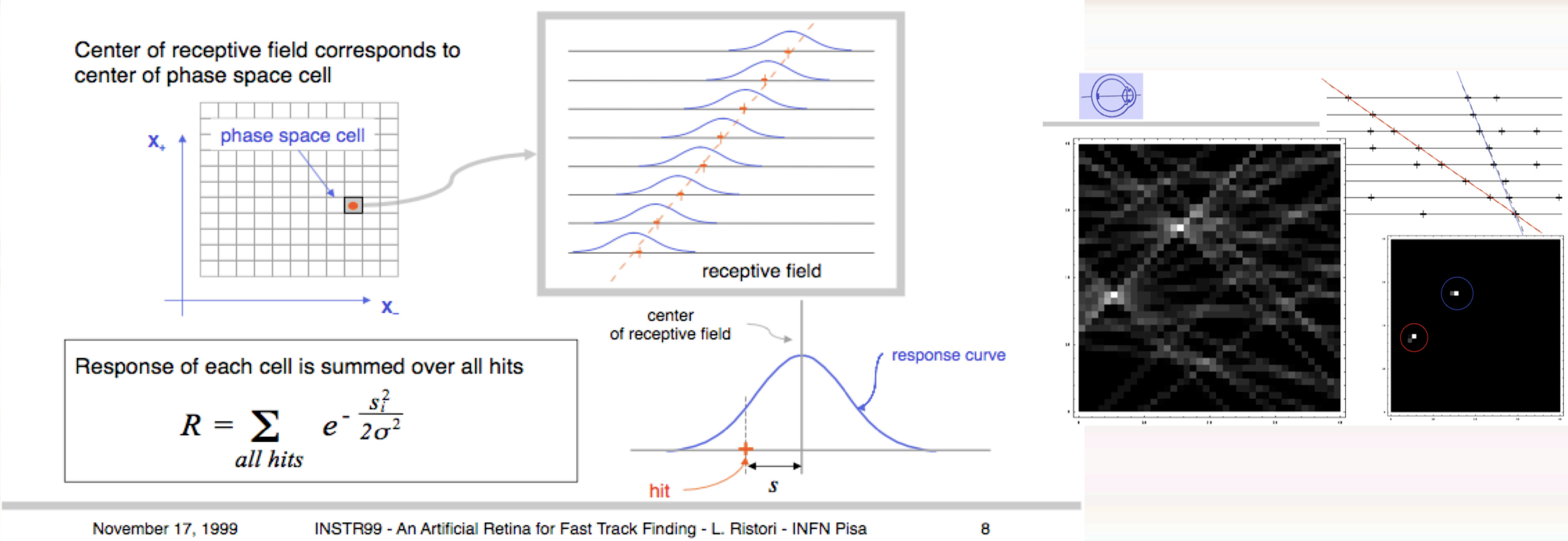
- Hit processing in AM still happens serially, while the visual system has no such serialization -> lots of processing power in the connectivity

- Second, the AM has “rigid templates”, while the brain works by interpolation of analog responses → this saves a lot of internal storage. Also, makes it easier to deal with “missing layers”.

- *Can we engineer these general concepts into a viable trigger system ?*

A "cellular" tracking algorithm

Inspired by mechanism of visual receptive fields [D.H. Hubel, T.N. Wiesel, J. Physiol. 148 (1959) 574],

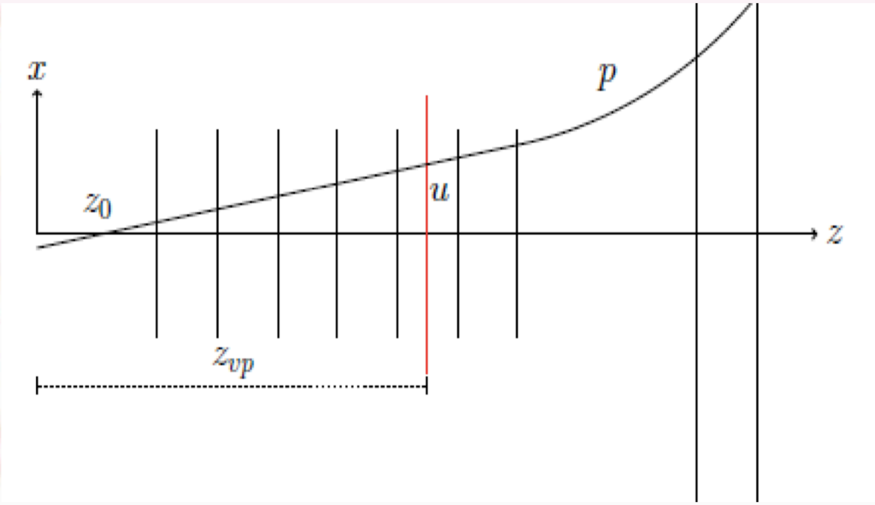


- Not really new: a study shown by one of us at INSTR99 showed that the idea is conceptually implementable in a toy tracker although not considered viable at the time of CDF SVT [NIM A453 (2000) 425-429]
- Vaguely related to "Hough transform" [P.V.C. Hough, Conf.Proc. C590914 (1959) 554]
- However, it takes ***a lot*** more to design an actually competitive system

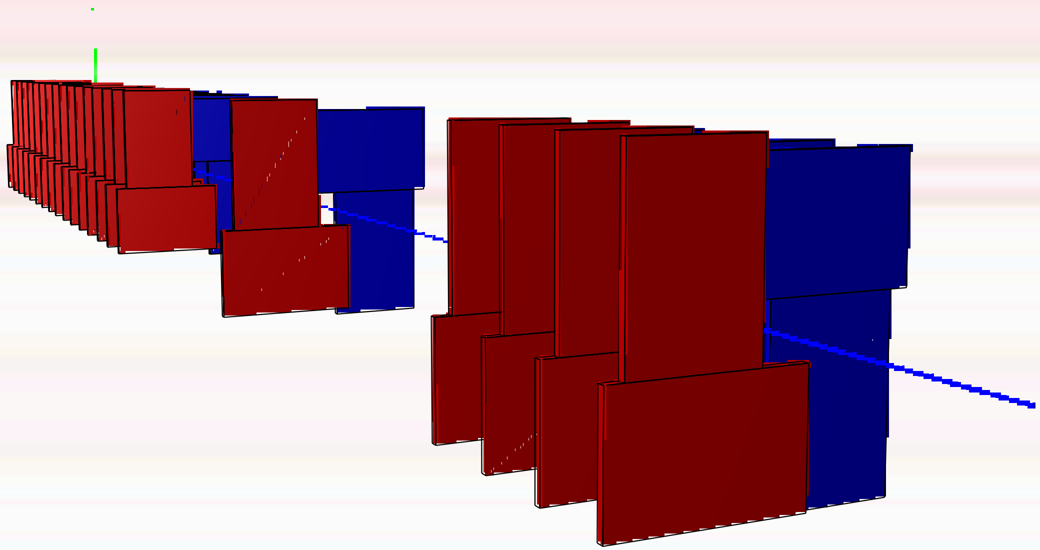
Today I describe a realistic implementation on a realistic pixel detector, with existing electronic components.

Geometry and track parameters

- An array of pixel detectors
- Each detector plane provides a (x,y) point at fixed z
- Measure straight tracks in 3D (4 parameters)
- e.g.: θ_x , θ_y , z_0, d (impact parameter)
- In case of presence of magnetic field, an additional pamater *p* is sufficient
- Does not need to assume B uniform, or perfect alignment



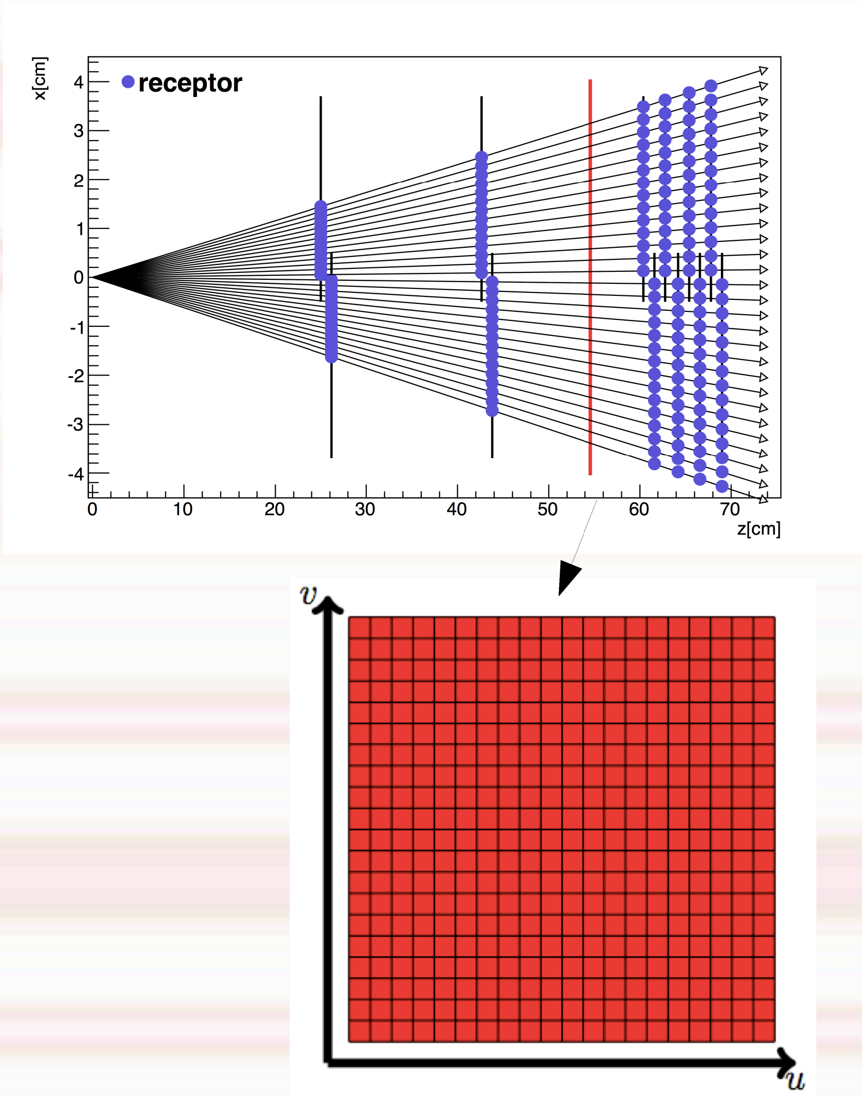
Realistic geometry example



- LHCb planned upgraded VELOPIX detector [LHCb-INT-2013-025)]
- Picked a 6-layer telescope for this exercise
- Neglect B field.

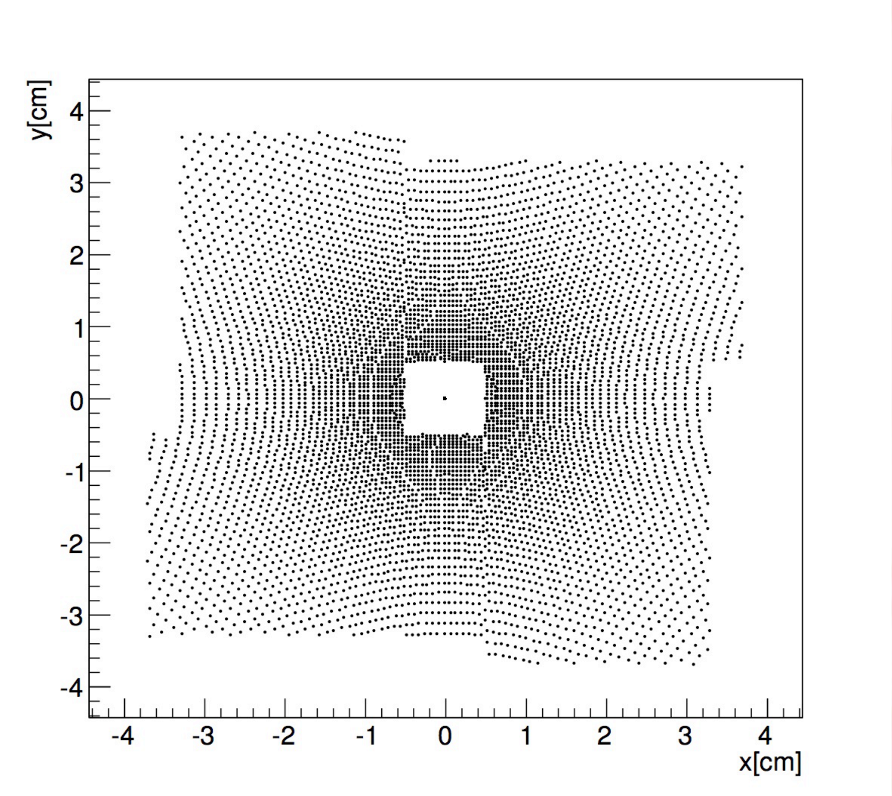
Mapping to detector to a receptor cell array

- Easy and intuitive way is to take two parameters from the intersection of tracks with an arbitrary plane
- This two parameters can then be mapped to a 2D main grid
- Remaining track parameters are implemented in 2 step

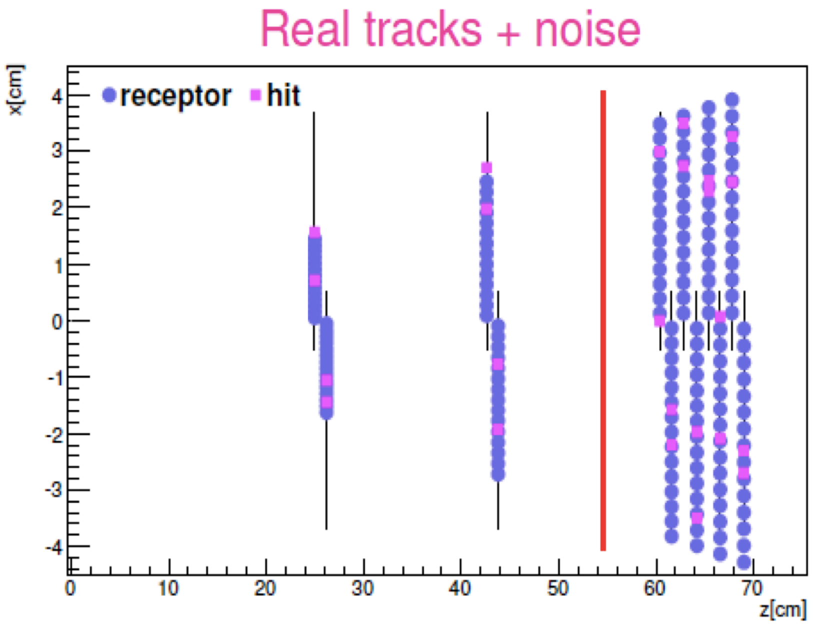


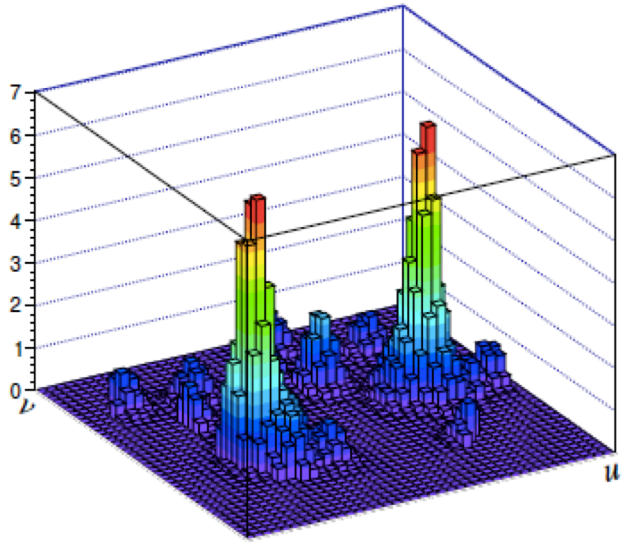
Mapping to detector to a receptor cell array

- Intersection of “base tracks” with detectors gives a map of “nerve endings”
- Every hit on the detector produces a signal on nearby receptors, depending on distance
- (I skip on several subtleties. For instance, effective operation require distribution to be non-uniform)
- (not unlike the distribution of photoreceptors in visual system – but it is all virtual in our case, that is, implemented in the electronic network connections)



Tracks appear as clusters in the cell array





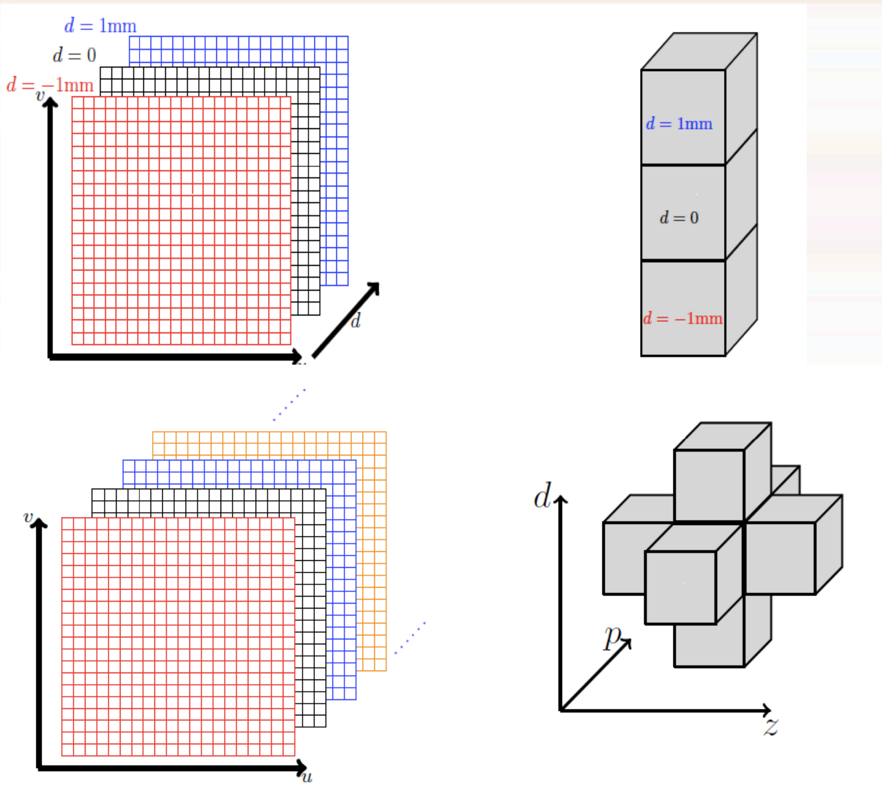
$$R_i = \sum_k e^{-s_{ir,k}^2/2\sigma^2}$$

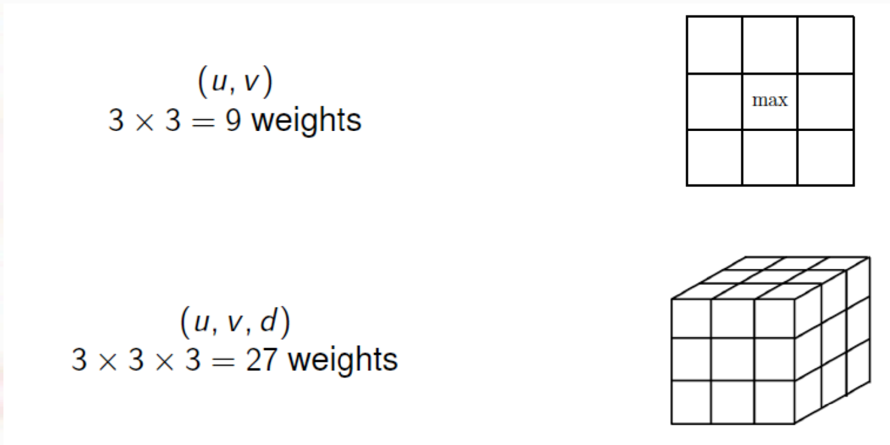
$$s_{ir,k}^2 = (x_i - x_r)_k^2 + (y_i - y_r)_k^2$$

Distance of hit to nearby receptors

Parameter extraction

- u,v parameters extracted directly from cluster centroid
- What about other 2 or 3 parameters ?
  - Add “lateral cells” and interpolate their response
    - Enough for a good estimate due to limited parameter spread.

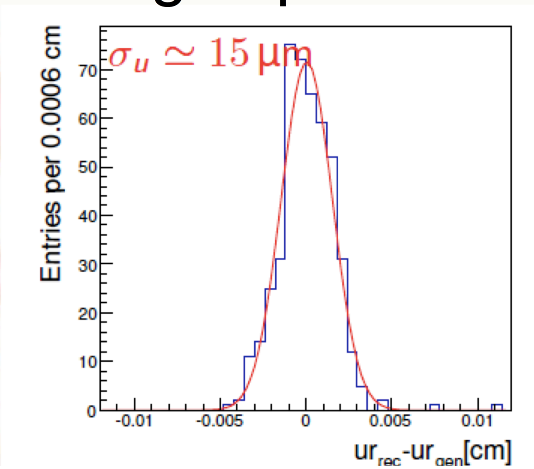


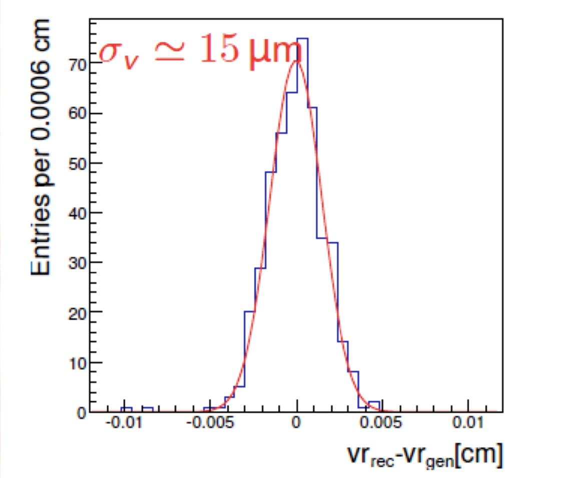


- We did the exercise up to 3^5=243 cells (full 5-parameters tracks)

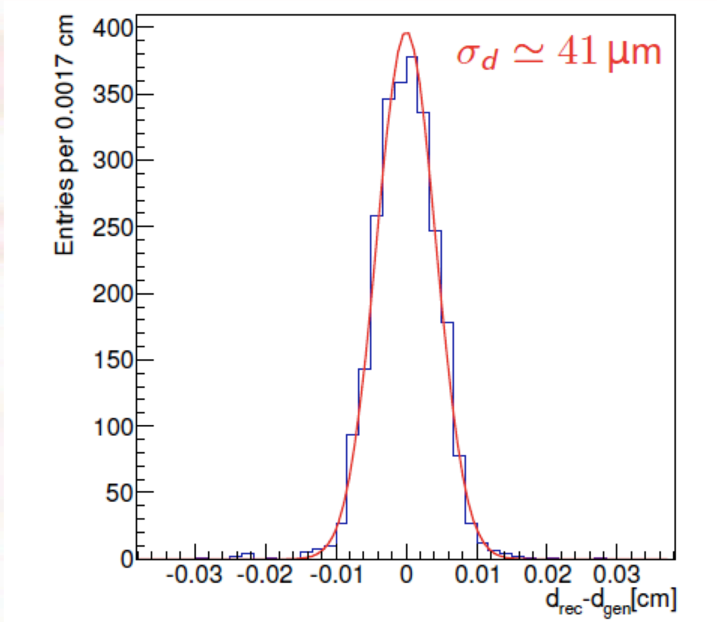
Results

Main grid parameters





Impact parameter resolution



(Compare with offline-style fit: σ = 40μm)

Simulation parameters:
- Pixel size 55μm, res.~12μm
- Hit eff. ~95%
- # of cells 50,000

- ***All Resolutions are offline-grade !***

Intermediate conclusions

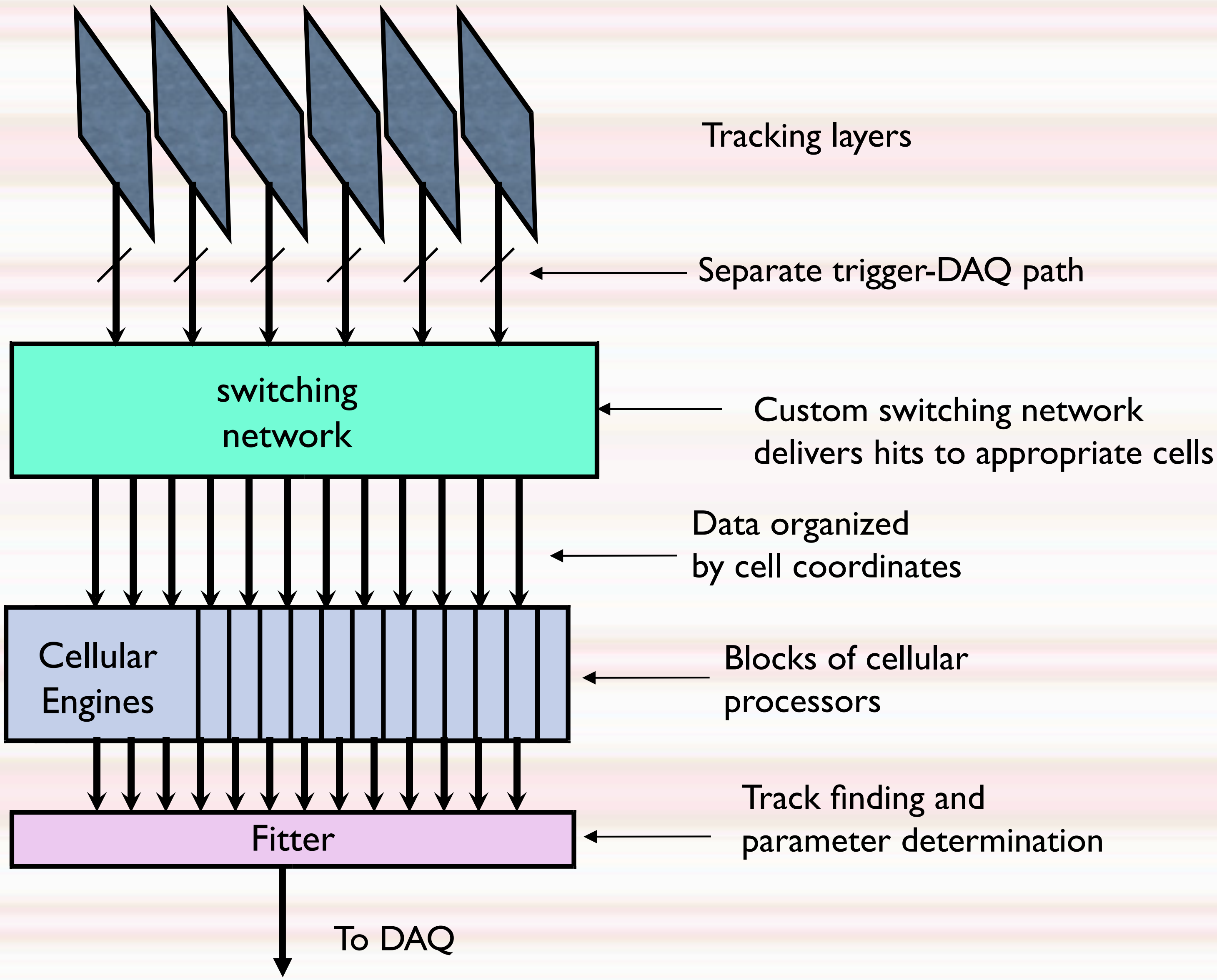
- We have shown with a realistic detector arrangement that It is possibile to reconstruct tracks and measure their parameters very well with a “brain inspired” cell-matrix method

- This algorithm is instrinsically very parallelizable

However:

Is it actually implementable in a hardware with reasonable size, cost, and with the needed timing to work at LHC crossing frequency ?

System Architecture

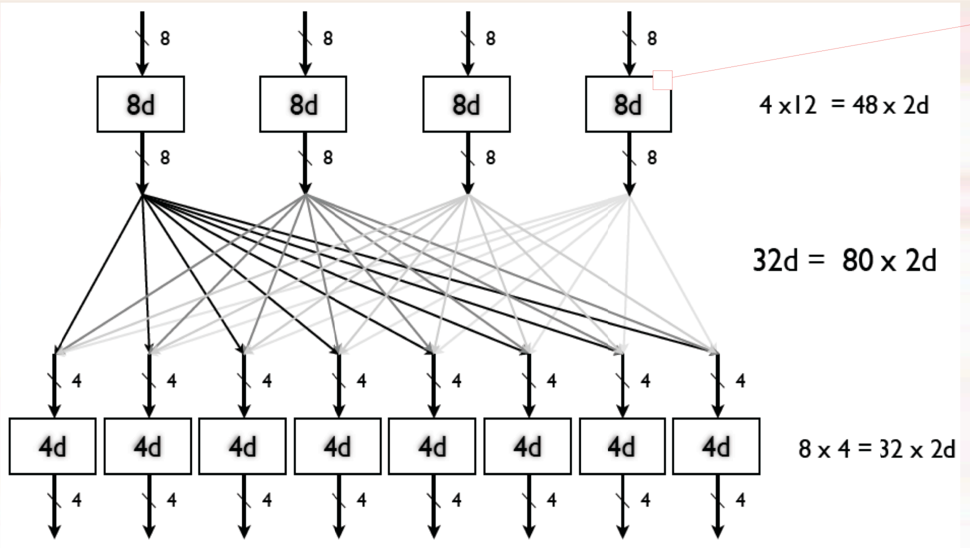


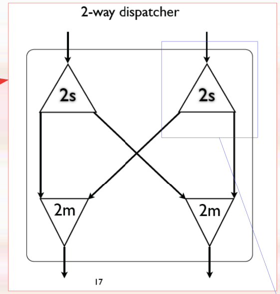
Implementation

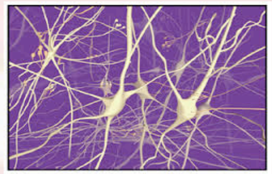
- Use modern, large FPGA devices.
  - Large I/O capabilities: now O(Tb/s) with optical links !
  - Large internal bandwidth – a must !
  - Fully flexible, easy to program and simulate
  - Steep Moore's slope, and easy to upgrade
  - Highly reliable, easy to maintain and update
  - Industry's method of choice for complex project with a small number of pieces (CT scanners, high-end radars...)
- We used Altera's Stratix V
  - Same device used elsewhere in LHCb readout system.

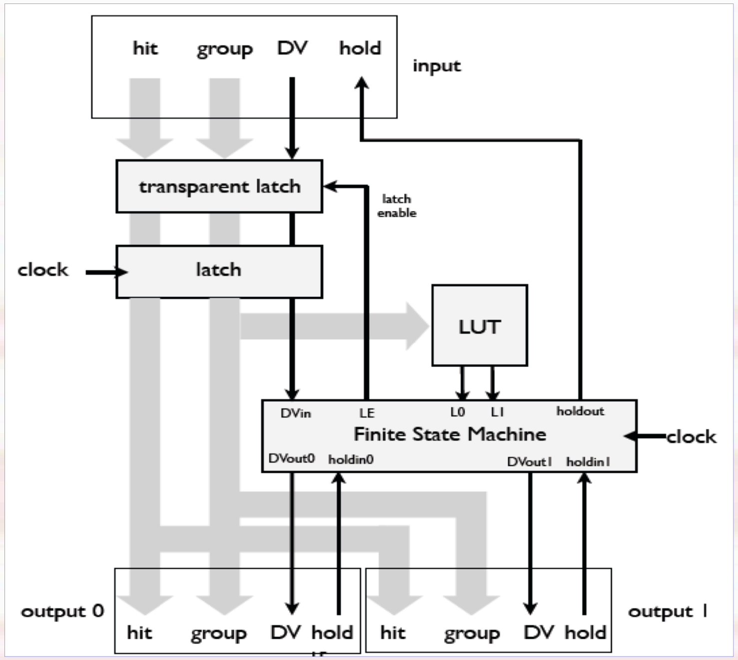
Let's find out if commercials say the truth....

Hit delivery by the switching logic



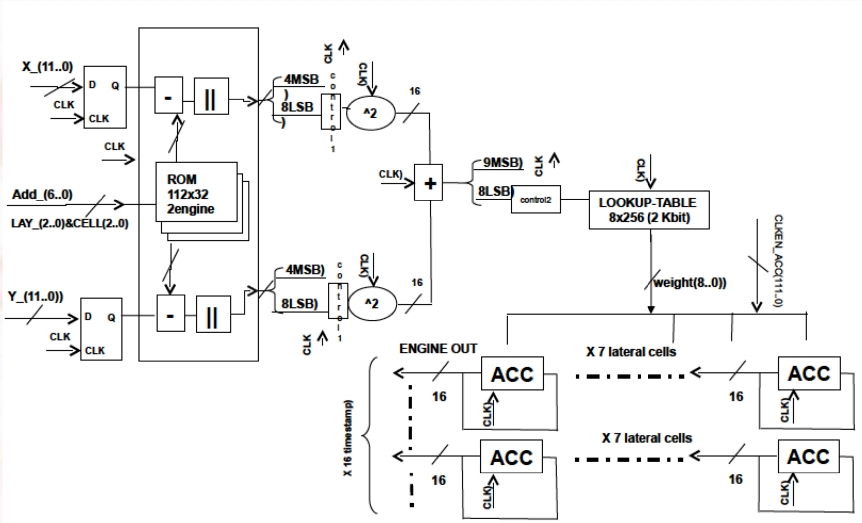




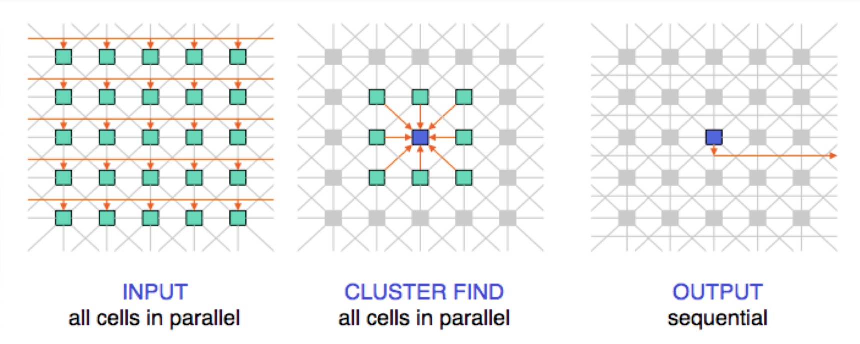


- Hits must be delivered only to the cell that need them (they can be more than one)
- The switch network “knows” where to deliver hits
- All information about the network of connections is embedded in the network via distributed LUTs

Cellular engine

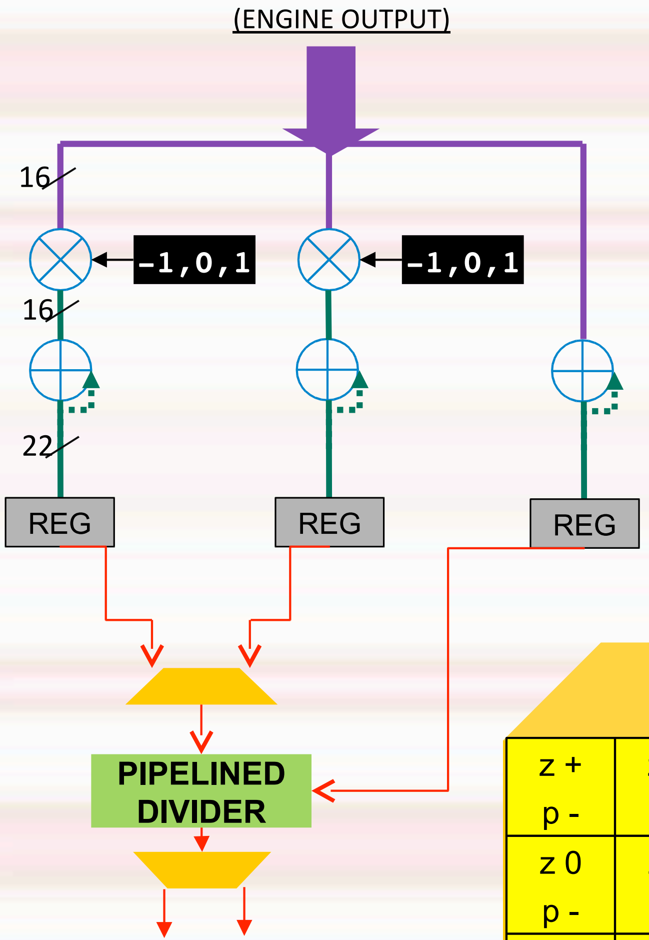


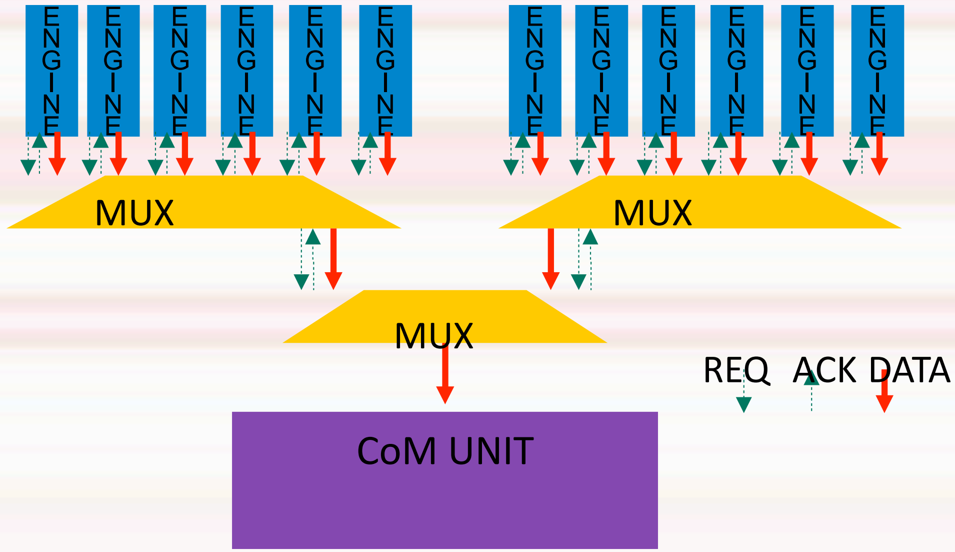
- Performs calculation of weights for a hit into a cell
- Deals with surrounding cells as well.
- Handles time-skew between events

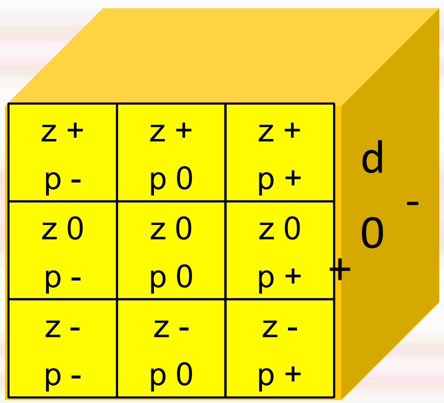


- In second stage performs lo-cal clustering in parallel, and queues results to output

Track parameter estimation by cluster Center-of-Mass







Due to data reduction out of the engine, a 1:12 ratio is sufficient to keep up with the data flow

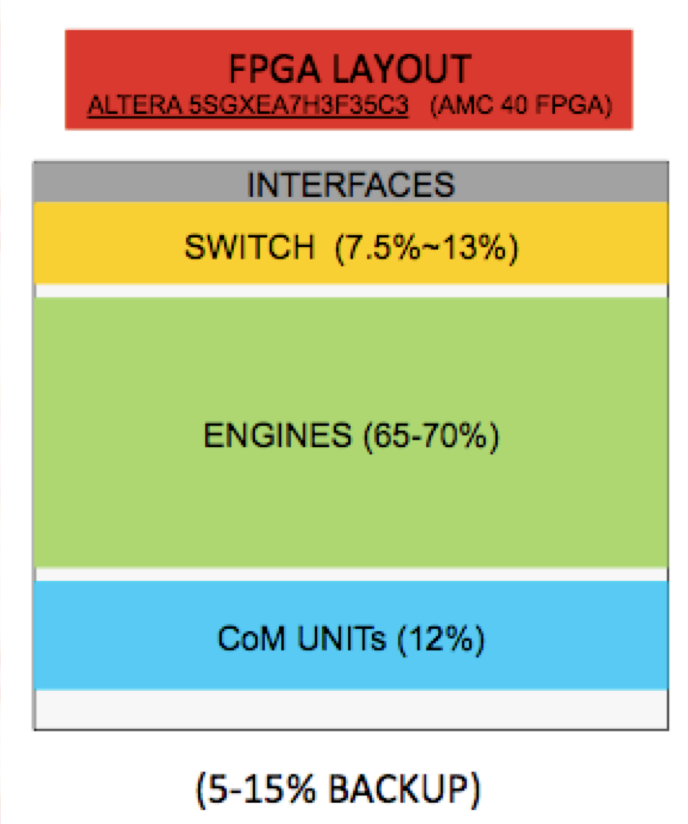
Fitting within a Stratix-V device

- All main components:
  - Switch
  - Engines
  - CoM

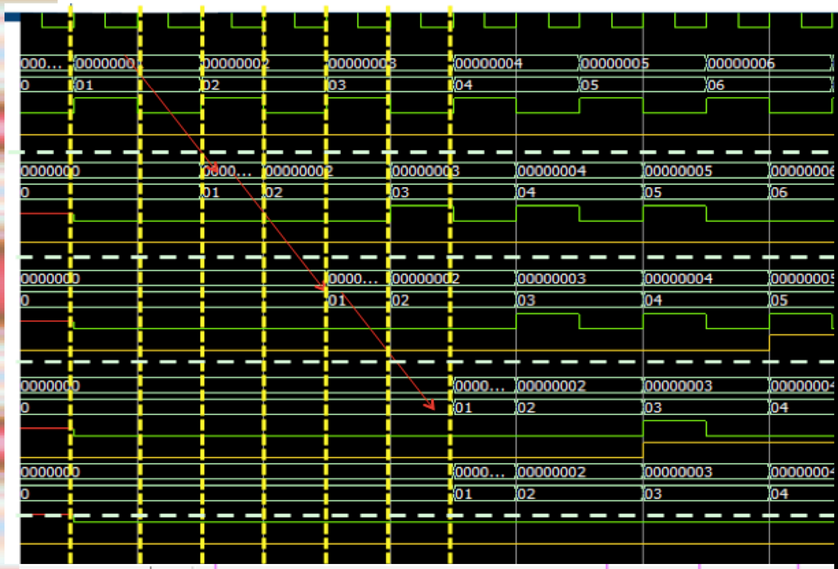
  implemented in VHDL and placed in the FPGA

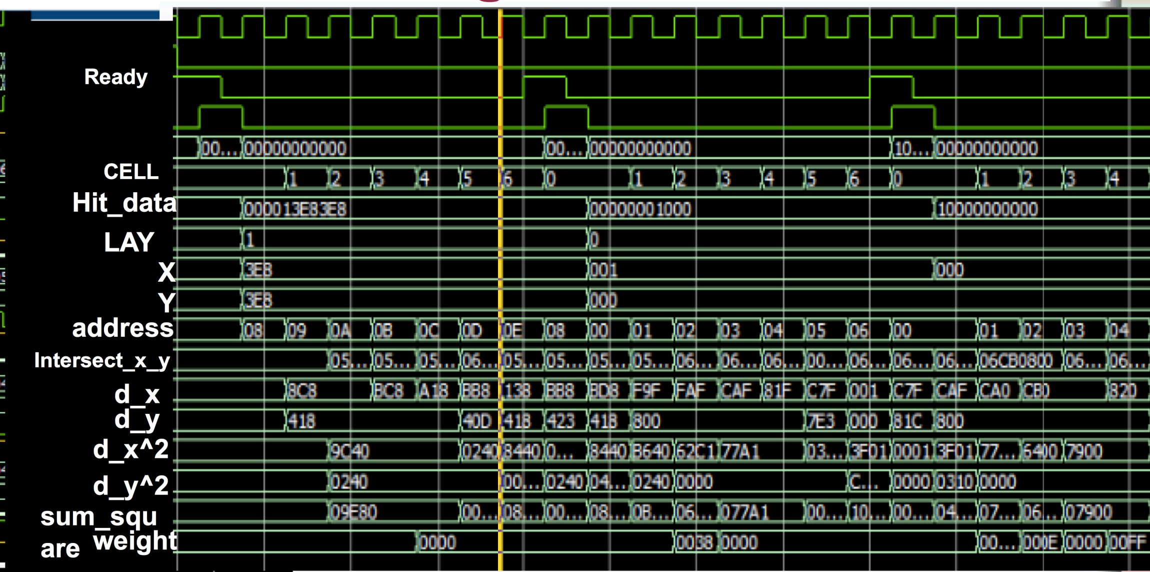
- Can fit O(10^3) engines/chip
  - exact number depends on details (time-ordering of pixel data, etc.)

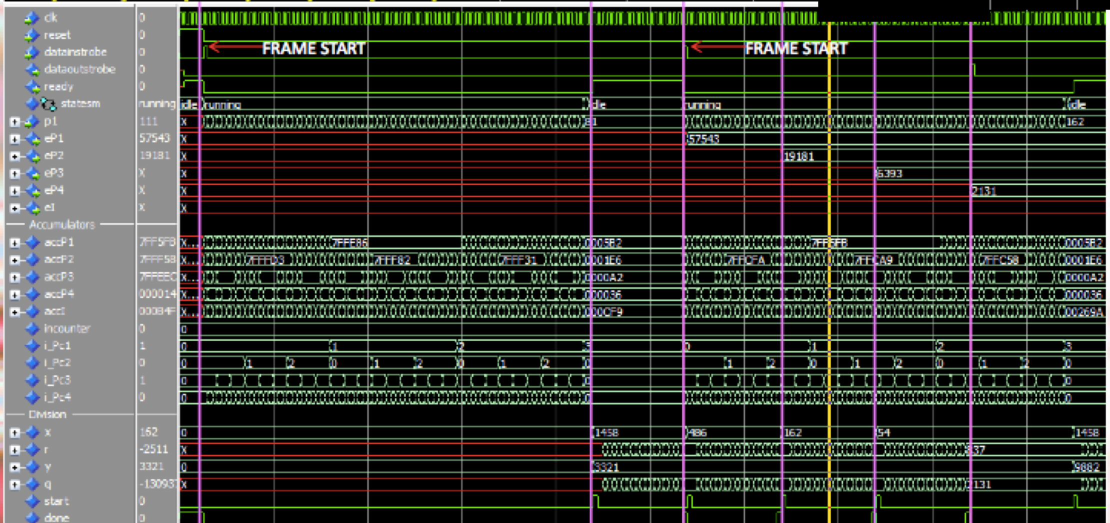
- Implies that a meaningful tracking system can be build with O(100) chips



Simulation and Timing







Exceed 350 MHz clock freq

40MHz throughput

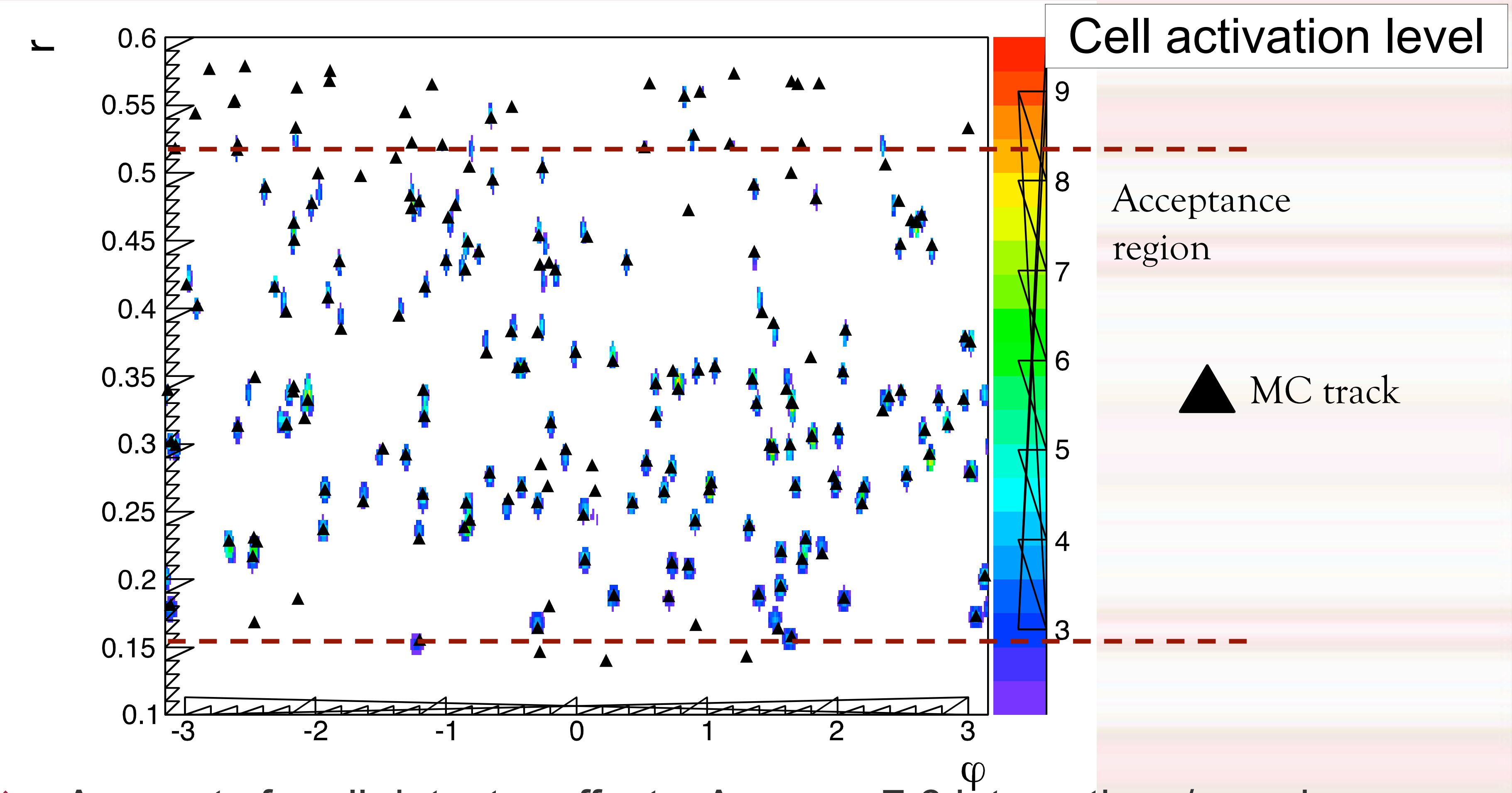
Total latency <1μs !

DISPATCHER 15 t.u.	FANOUT 6 t.u.	ENGINES 70 t.u.	CoM 55 t.u.	FINAL 2-10 t.u.

~ 156 t.u. = 436 ns

(Not accounting for I/O)

Further progress: LHCb full-MC at upgrade luminosity



- Accounts for all detector effects. Average 7.6 interactions/crossing.
- Recently extended with addition of B field and further tracking layers (UT)
- Efficiency/ghost rate performance comparable to offline reconstruction.

CONCLUSIONS

- We showed that the “retina algorithm” actually allows real-time track reconstruction in a real HEP detector application.

- We developed a design for a real-time track processor that works at LHC crossing frequency, with latency ~1μs
  - Specific R&D for LHCb already well advanced

- Enpowers experiments at high-luminosities to work as if reading complete tracks straight out of the detector. Might lead to fruitful future developments.