Status report on Retina activities in Milano

Marco Petruzzo
Università degli Studi and INFN, Milano

Outline

- **Artificial retina algorithm**
- **2D tracking prototype and testbeam results**
- **3D/4D Artificial Retina simulations**
- **Artificial Retina implementation on gFEX board**
- **Conclusions**

Artificial retina algorithm for real-time tracking

Inspired from **neurobiology**:

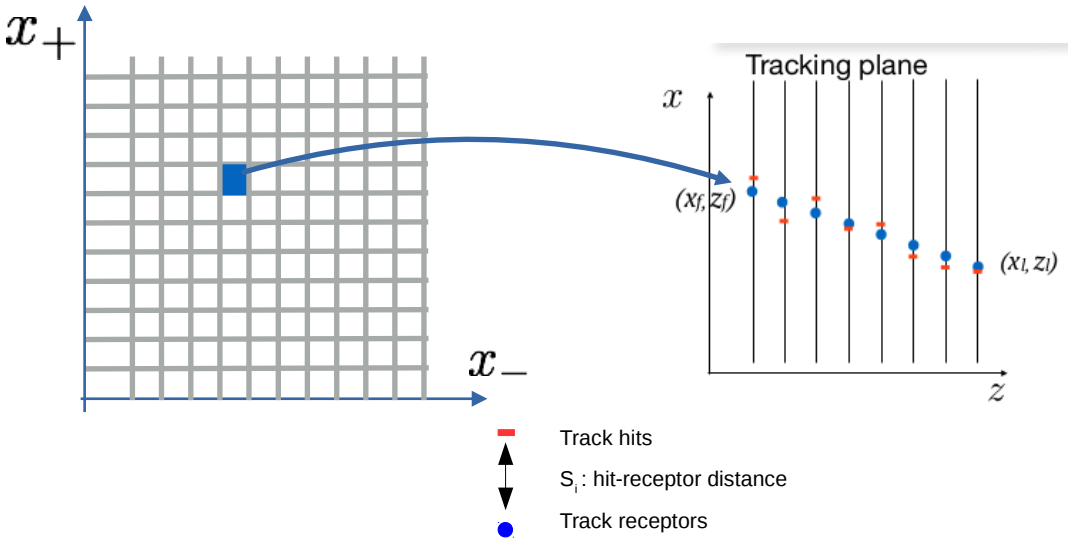
specific neurons of the retina are specialized **to identify specific shapes**

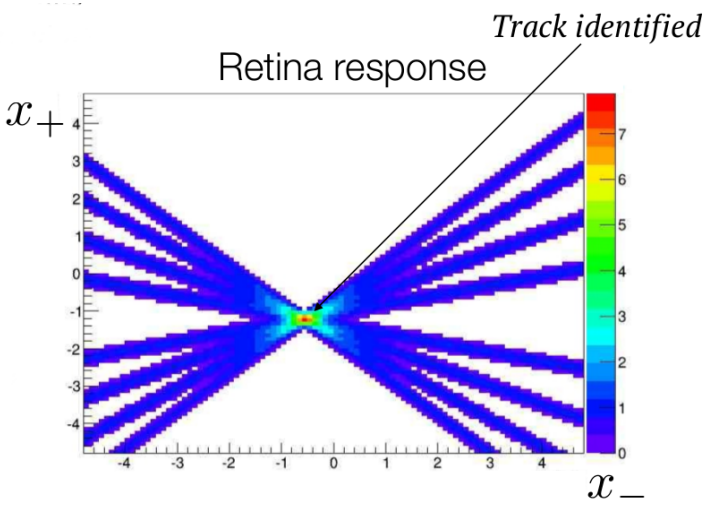
Can be applied to **tracking:**

A pool of **cellular units** (engines) **tuned to identify specific tracks**

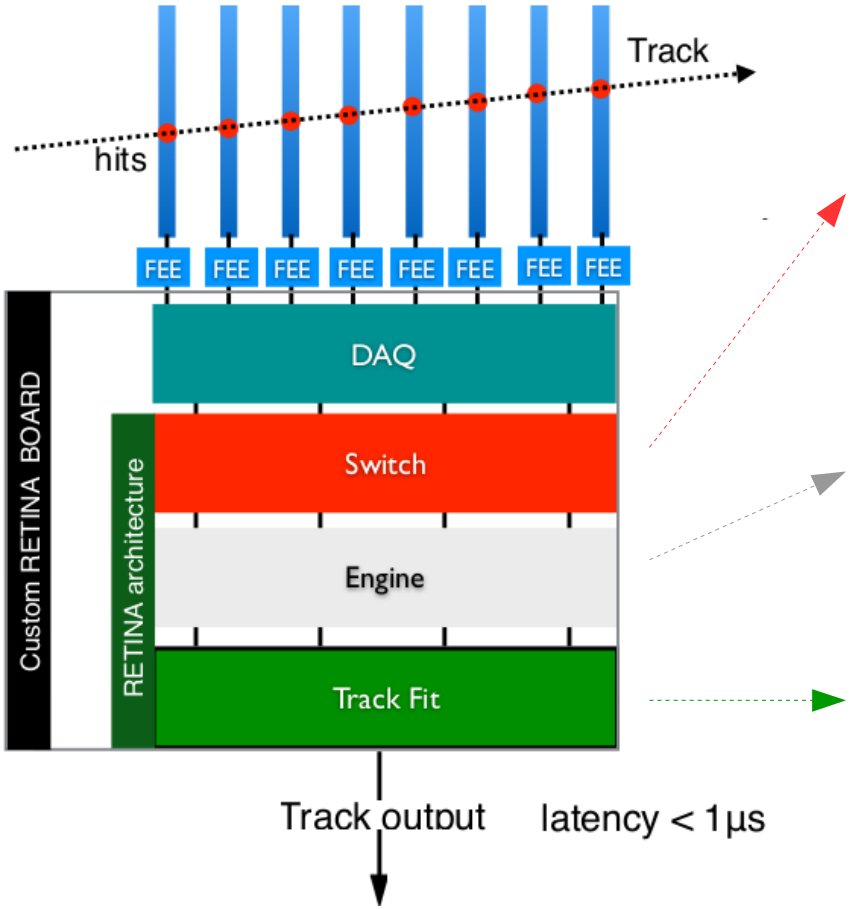
An engine is associated to a **precomputed track**

- Each detector **hit produces a stimulus to the engine**
- The response is proportional to **how close the hit is to the precomputed track**
- **Engines with maximum** excitation correspond to track candidates
- Track **parameters are obtained via interpolation** of the response near the maximum





Artificial Retina architecture



Switch:
delivers the hits to the engines with non-negligible response

Engines:
evaluate the response to the incoming hits for different track hypotheses

Track Fitter:
identifies and **interpolate the local maxima** of the response and outputs the result to disk

- **Highly parallelized** algorithm
- **Pipelined** architecture
- Particularly suitable for **implementation in FPGA**

- **Offline-like** quality tracks with **sub-µs latencies**
- **Track information** available in **real time**
- can be used in the **L1 trigger**

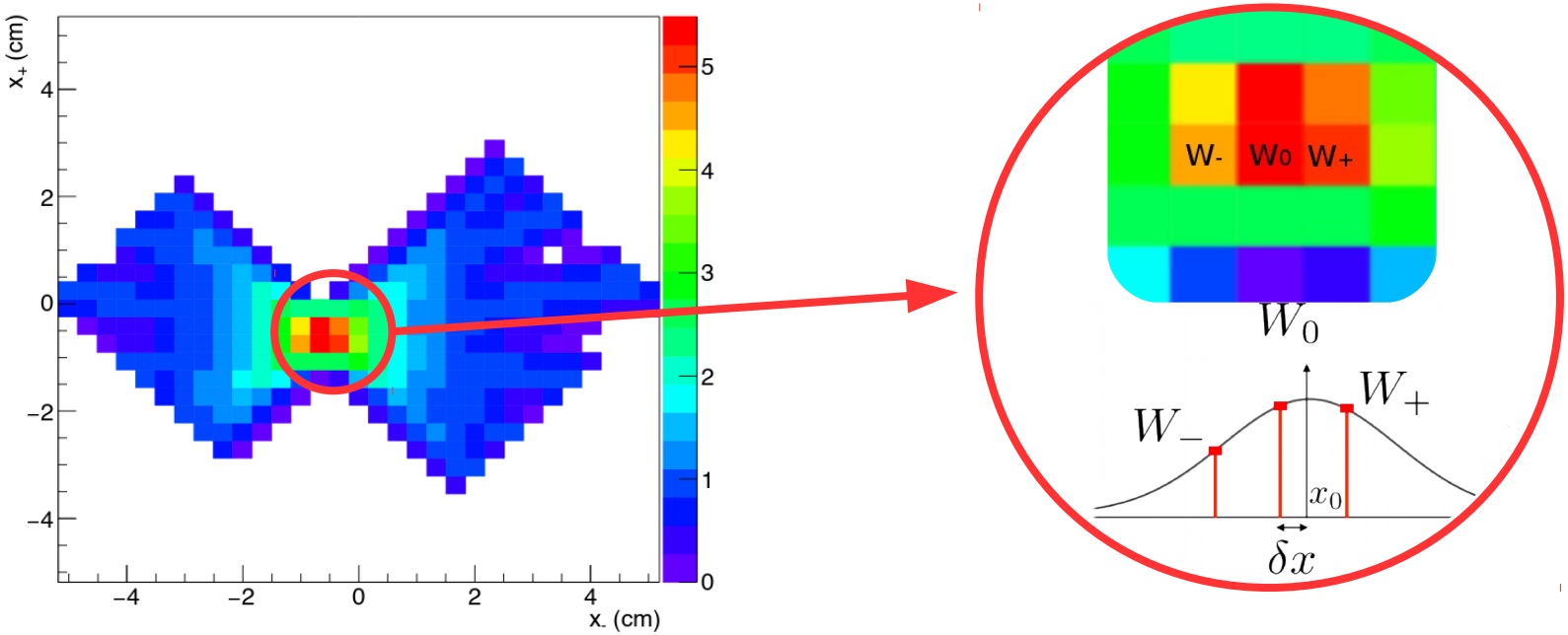
First real-time 2D tracking prototype

The Artificial Retina has been **implemented in Xilinx Kintex 7 FPGA and proved to work** for a prototype system based **on a silicon strip telescope.**

Artificial Retina algorithm for 2D tracking:

- **512 cellular units** within the acceptance
- Gaussian response to the **hit-track distance** s_{ijk}

$$W_{ij} = \sum_k \exp\left(-\frac{s_{ijk}^2}{2\sigma^2}\right)$$



- Track parameters evaluated via i**nterpolation of the response** near the identified maximum in (i,j)

$$x_{+,\mathrm{rec}} = x_{i+} + \frac{\Delta_{x_+}}{2}\frac{\ln(W_{i-1\,j}/W_{ij}) - \ln(W_{i+1\,j}/W_{ij})}{\ln(W_{i-1\,j}/W_{ij}) + \ln(W_{i+1\,j}/W_{ij})},$$

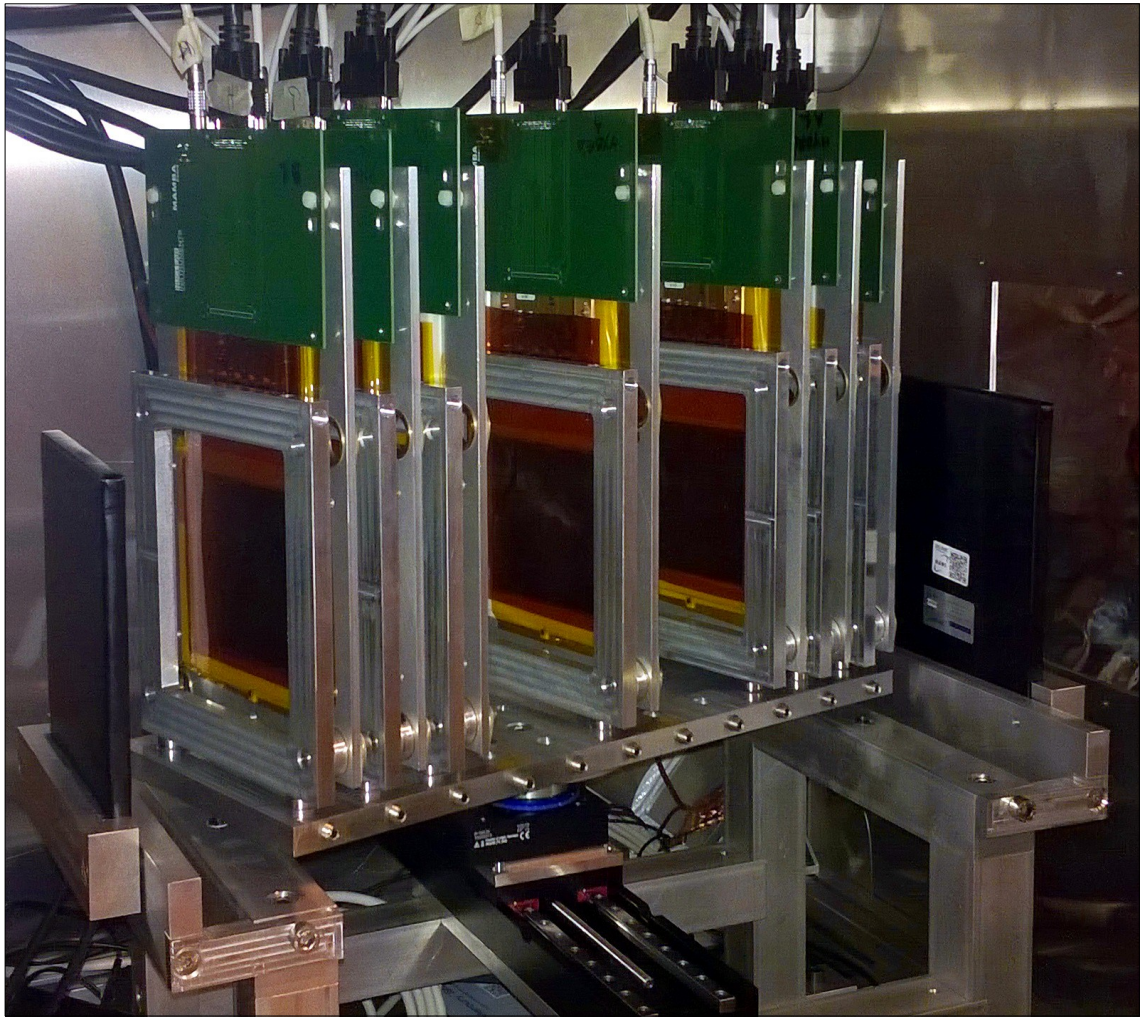
$$x_{-,\mathrm{rec}} = x_{j-} + \frac{\Delta_{x_-}}{2}\frac{\ln(W_{ij-1}/W_{ij}) - \ln(W_{ij+1}/W_{ij})}{\ln(W_{ij-1}/W_{ij}) + \ln(W_{ij+1}/W_{ij})}$$

2D tracking prototype at SPS, CERN

A **silicon strip telescope** has been developed and produced for testing the Artificial Retina.

Test setup at SPS, CERN:

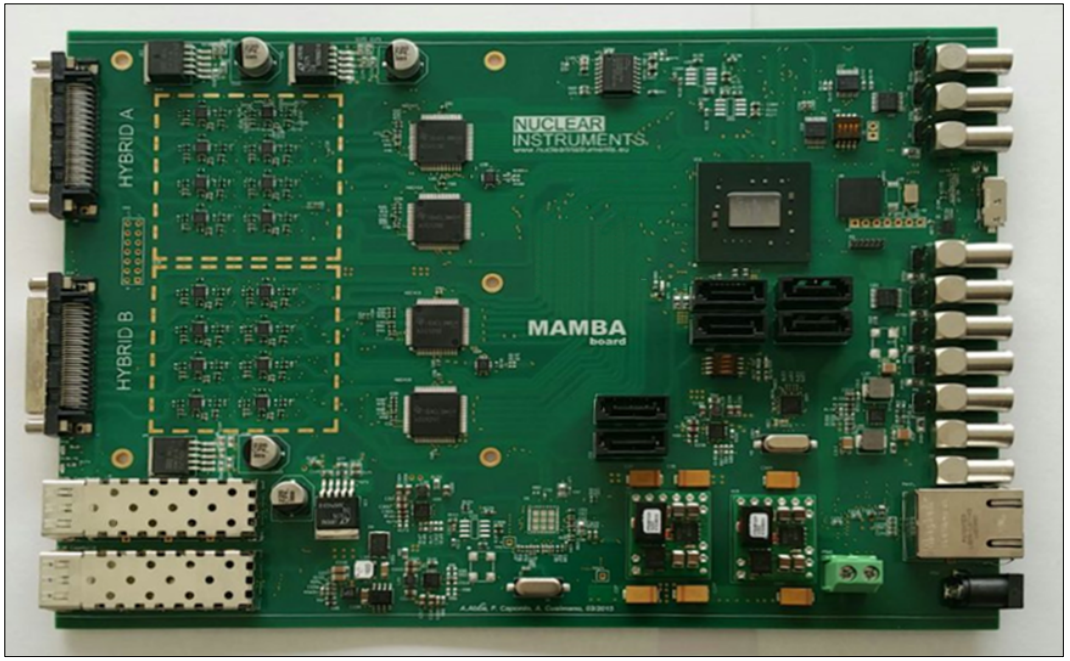
- **7 single-sided strip sensors** (STM OB2 sensors):
  - ~10x10cm^2 active area,
  - 512 strips, 183μm pitch
- **Two plastic scintillators** as trigger
- **Linear and rotation stage** used to move the telescope inside the dark box
- Telescope enclosed in a dark box with cooling system.
- **DAQ and Artificial Retina** implemented on a custom board



MAMBA DAQ+RETINA board

MAMBA board:

- Milano Advanced Multi Beetle Acquisition board
- based on Xilinx Kintex 7 FPGA
- up to 8 planes readout at 300KHz (max. Beetle chip readout rate)
- 12bit ADCs for Beelte signals digitalization
- on-board Artificial Retina algorithm



Other “DAQ only” purposes:

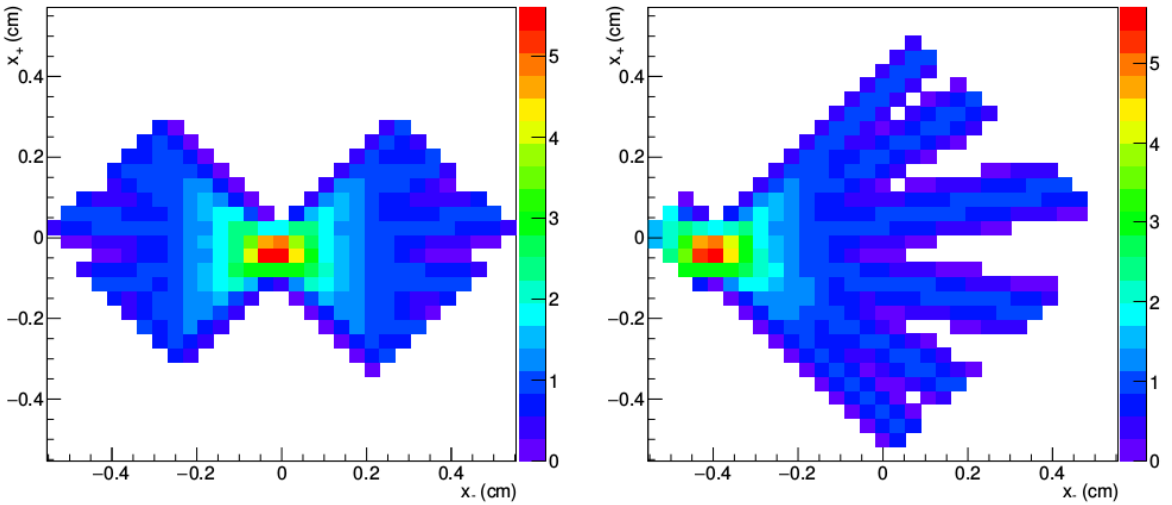
- Readout of a DUT in the UT-LHCb testbeams in 2015-2016
- Readout of the silicon telescope for DUT studies using cosmic rays

Testbeam results

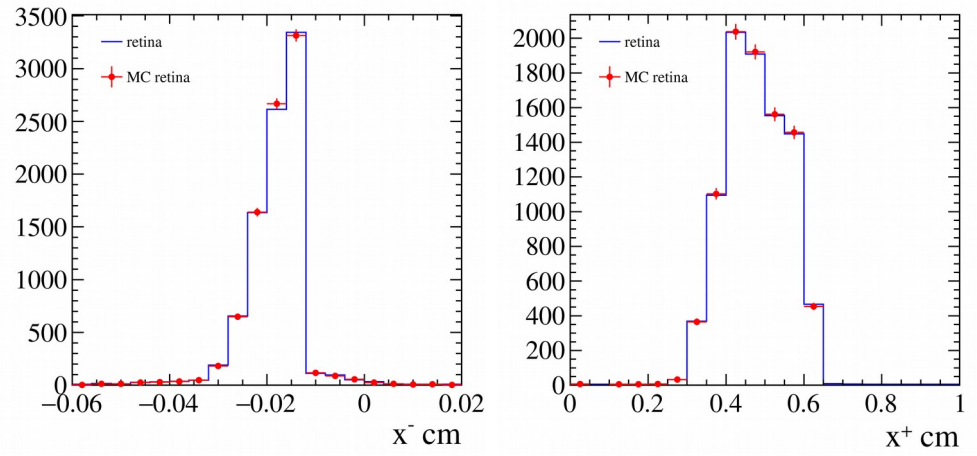
The Artificial Retina algorithm has been tested for track impinging at **different angles and positions.**

Typical response of the Artificial Retina for 1-track events:

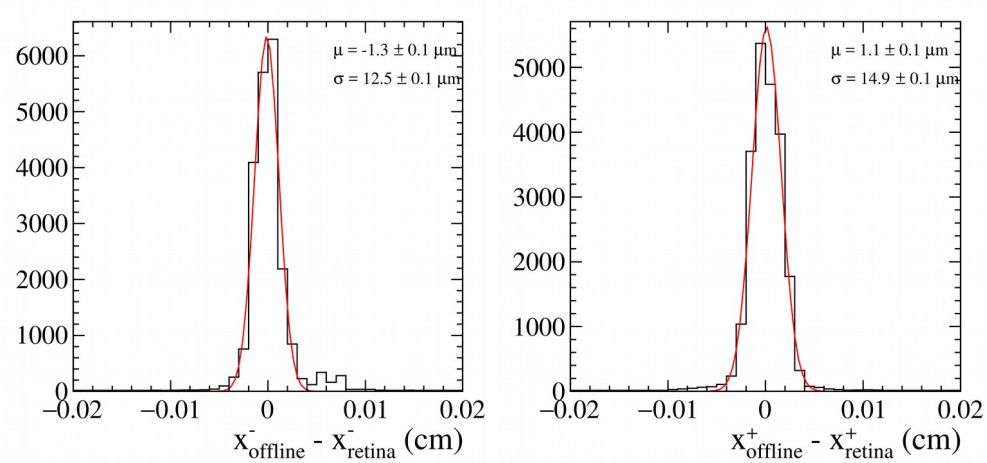
- Left: 0° track angle
- Right: 20° track angle



Track parameters distribution determined by the artificial retina.Testbeam data processed by the MAMBA board (retina) compared to the artificial retina simulated response (MC retina)



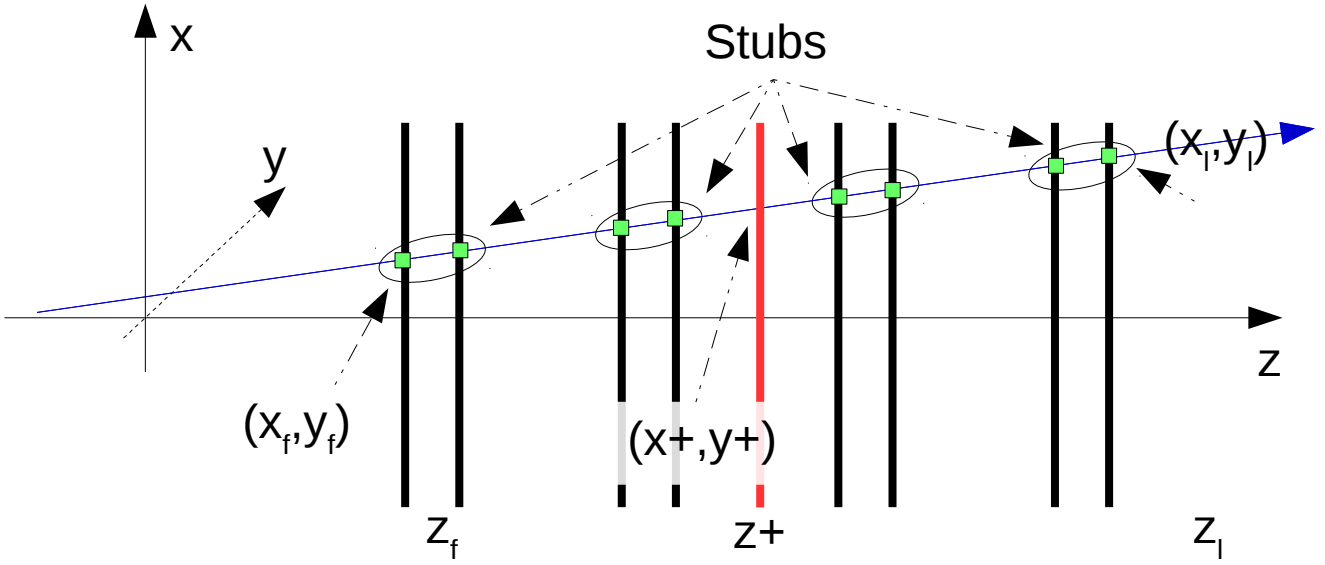
Distribution of the residuals for the track parameters evaluated using a simple χ^2 minimization algorithm (offline) and track parameters from the artificial retina algorithm



4D tracking at HL-LHC - Simulations

VELO-like tracking device:

- 12 planes of silicon pixel detectors in the forward region
- 60x60mm^2 sensor size
- 55x55μm^2 pixel size
- 30ps time resolution



“Stub” approach:

- Planes are considered **as couples**
- Stubs are constructed **linking hits** from adjacent planes
- **Cuts are applied** based on the spatial parameters of the stub
- **Velocity** is required to be compatible with the **speed of light**

Definitions:

$(x_f, y_f, z_f), (x_l, y_l, z_l)$

: intersections of the track with first and last tracking plane

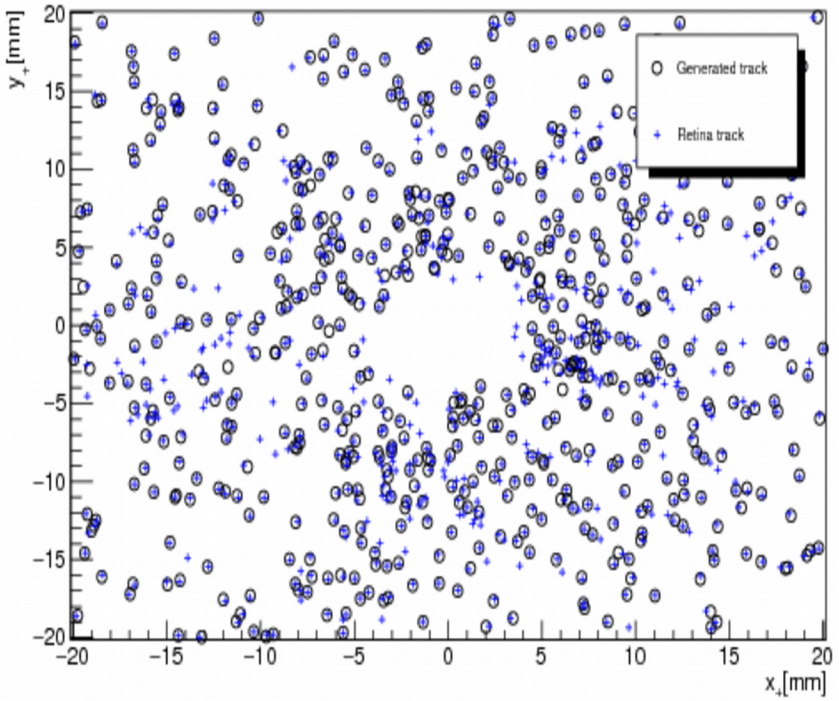
$$x_{\pm} = (x_f \pm x_l)/2$$
$$y_{\pm} = (y_f \pm y_l)/2$$
$$z_{\pm} = (z_f \pm z_l)/2$$

: (x+,y+) are the intersections with a tracking plane placed at z+
: (x-,y-) define the slope of the track

Retina response and track parms. evaluation

Stub space-time parameters:

$$\begin{pmatrix} x_- \\ x_+ \\ y_- \\ y_+ \\ t \end{pmatrix}_{stub} = \begin{pmatrix} \frac{x_1 z_- - x_2 z_-}{z_1 - z_2} \\ \frac{x_1 (z_+ - z_2) - x_2 (z_+ - z_1)}{z_1 - z_2} \\ \frac{y_1 z_- - y_2 z_-}{z_1 - z_2} \\ \frac{y_1 (z_+ - z_2) - y_2 (z_+ - z_1)}{z_1 - z_2} \\ \frac{t_1 + t_2}{2} - \frac{z_1 + z_2}{2\, c\, \sqrt{1 + (x_-/z_-)^2 + (y_-/z_-)^2}} \end{pmatrix}$$



Response to one stub:

- Distance between the (i,j) engine and the k[th] stub distance

$$s^2_{ijk} = (x_{k+} - x_{i+})^2 + (y_{k+} - y_{j+})^2$$

- The k[th] stub produces a Gaussian excitation to the engine

$$W_{ijk} = \begin{cases} \exp\left(-\frac{s^2_{ijk}}{2\sigma^2}\right) & \text{if } s_{ijk} < 2\sigma \\ 0 & \text{otherwise} \end{cases}$$

Track parameters evaluation

$$W_{ij} = \frac{1}{N_{ij}} \sum_k^{N_{ij}} W_{ijk} \longrightarrow (x+,y+)_{trk} \text{ via Gaussian interpolation}$$

$$x_{-ij} = \frac{1}{N_{ij}} \sum_k^{N_{ij}} x_{-ijk}$$

$$y_{-ij} = \frac{1}{N_{ij}} \sum_k^{N_{ij}} y_{-ijk} \longrightarrow (x-,y-,t)_{trk} \text{ via average of the stub contributions}$$

$$t_{ij} = \frac{1}{N_{ij}} \sum_k^{N_{ij}} t_{ijk}$$

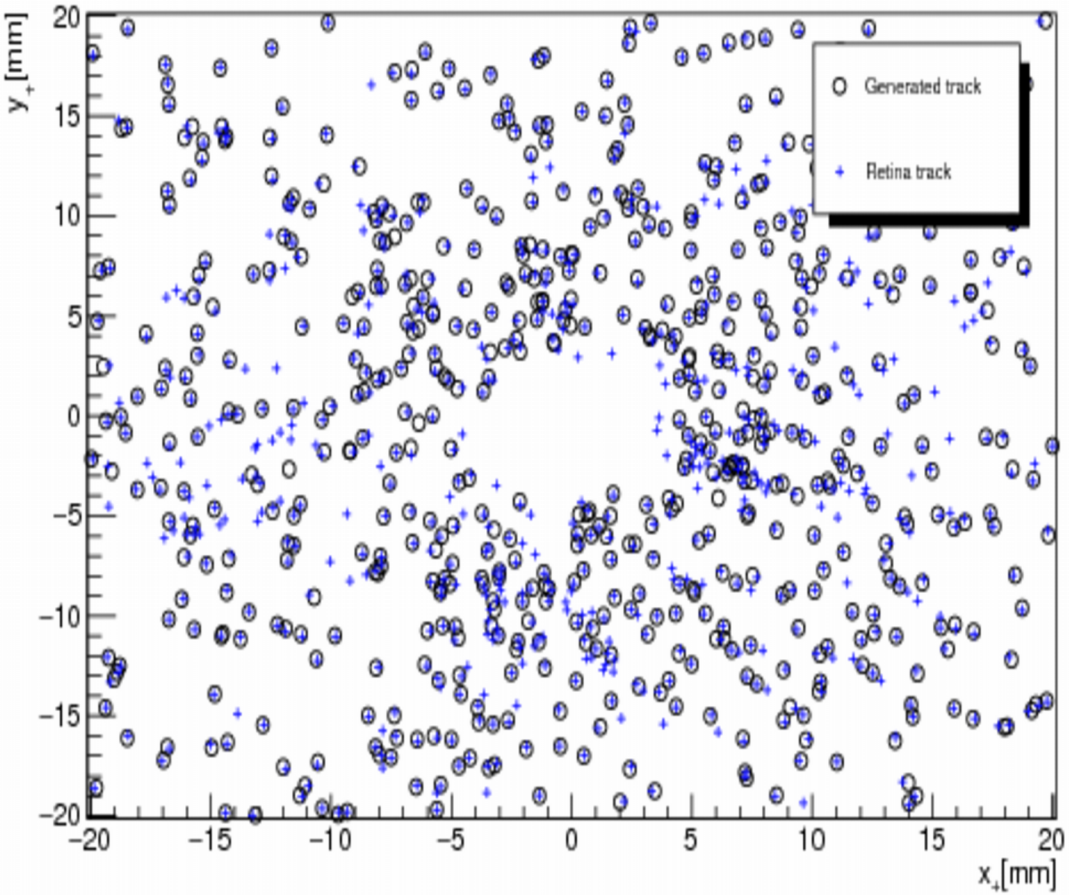
Simulation of the retina response

Track conditions:

- 1200 generated tracks/event (~600 within the retina acceptance)
- Interaction point Gaussian distributed in time and along the z axis: σ_z=5cm, σ_t=167ps

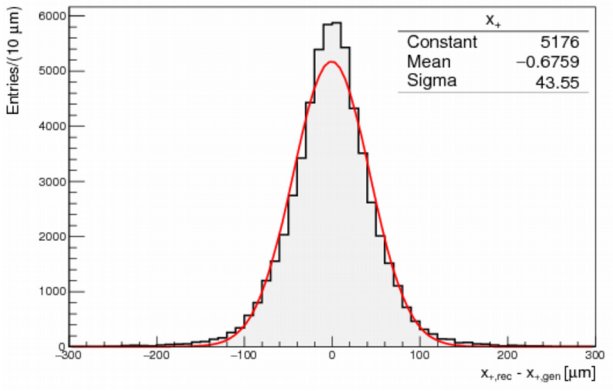
Engines distribution:

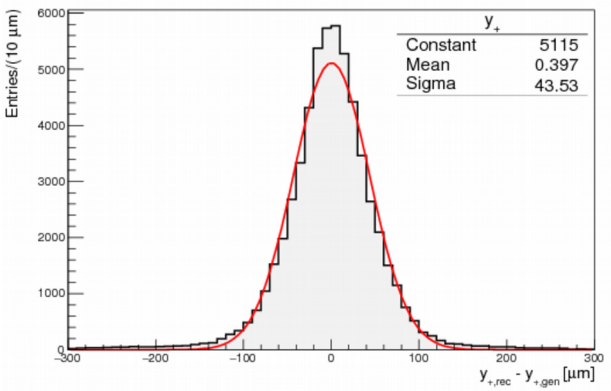
- 90’000 engines (~5000eng/FPGA)
- Uniform distribution in the (x+,y+) space : [-2,2]x[-2,2]cm^2 square
- Simulations with and without using the time information of the stubs



Retina response using time information

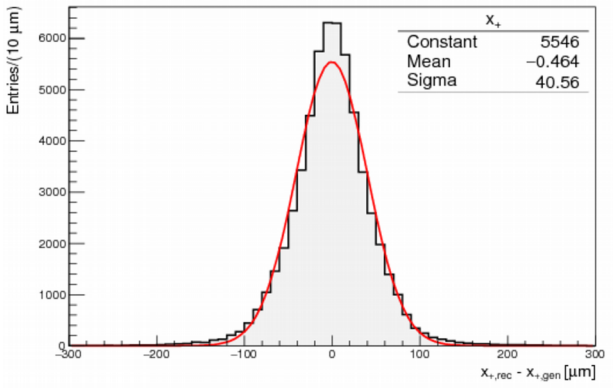
Simulation results

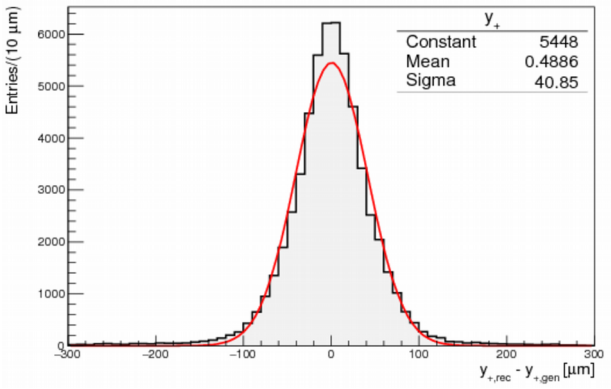


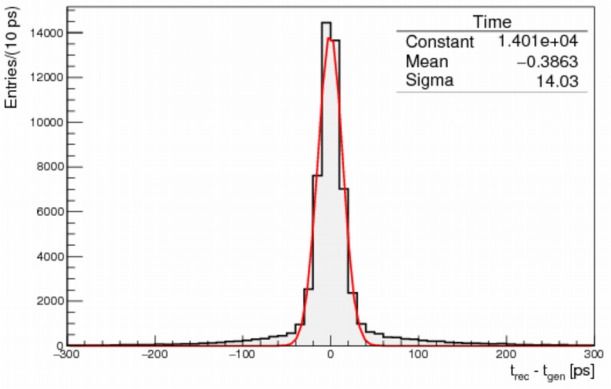


First row: resolution on track parameters (x+,y+) **without** using the time information of the stubs

Second row: resolution on track parameters (x+,y+, t) **using the time information** of the stubs







Tracking performance improves when including the time information

$\sigma_{x-/y-}$ = 95.0µm
$\sigma_{x+/y+}$ = 43.5µm

Time

$\sigma_{x-/y-}$ = 70.1µm
$\sigma_{x+/y+}$ = 40.6µm
σ_{t} = 14.3ps

The reconstruction efficiency is stable. The tracks **purity improves**.

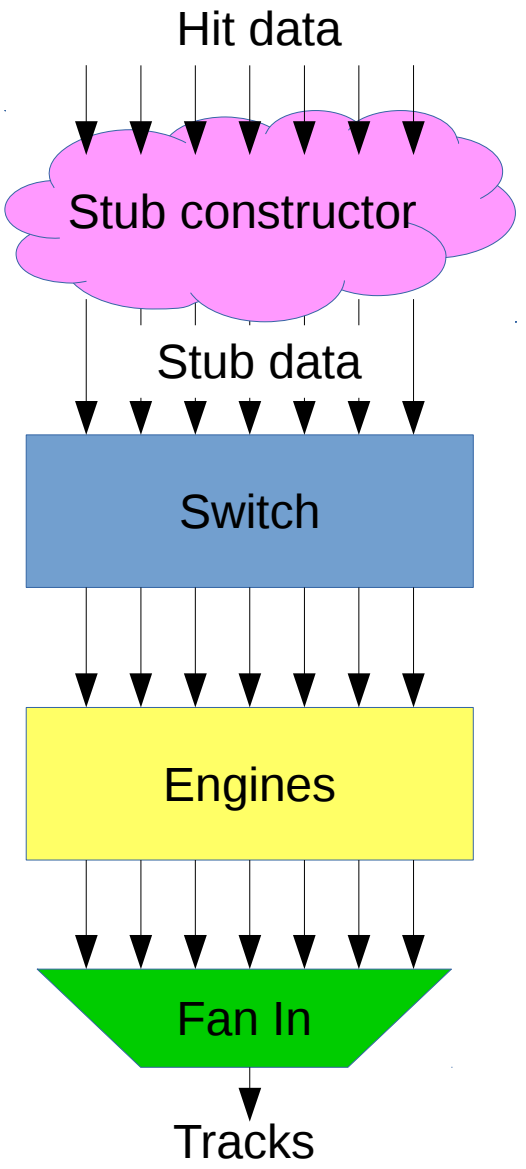
Efficiency = 99%
Purity = 64%

Time

Efficiency= 99%
Purity = 85%

Status of 3D/4D artificial retina implementation

- The stub constructor has to be designed and implemented.
- The switch is implemented using networks data mergers and dispatchers. Different modules have been designed and allows flexibility in the configuration of the switch according to the number of inputs and outputs.
- Engines are organized in “regions”, without lateral communication between regions nor engines inside the same region. (Communication between first neighbors can still be considered to find the local maxima of the retina response).
- The fan-in is used to collect the track data and deliver to the next stage of processing (outside the artificial retina). It can also be a switch, depending to the desired number of outputs.



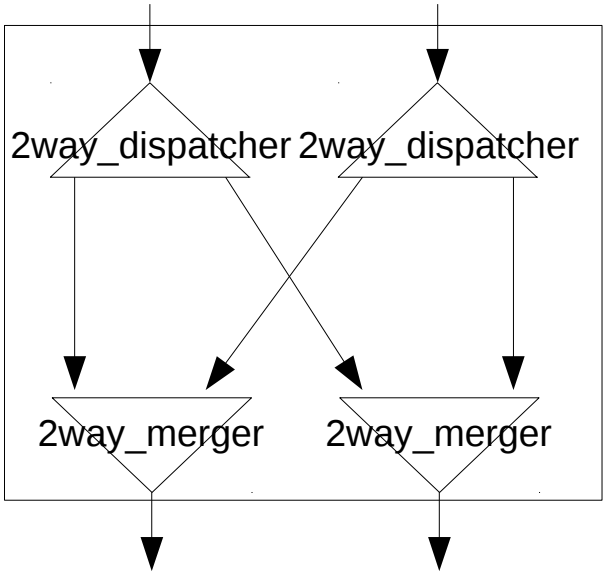
Switch implementation (1)

The Switch delivers the hits to the engines with non-negligible response, **based on a precomputed address:**

The simplest switch has **2 inputs, 2 outputs**, built using:
- **Two 2way_dispatchers**
- **Two 2way_mergers**

The 2way_dispatchers read the input data and delivers to one or the other output port according to 1 address bit.

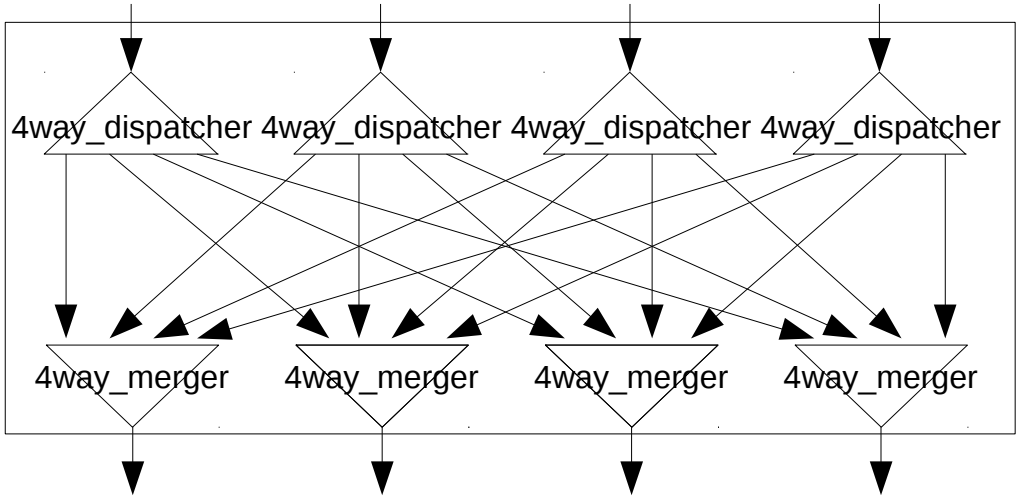
The 2way_mergers receives data from two inputs and manage the data flow to the only output port.



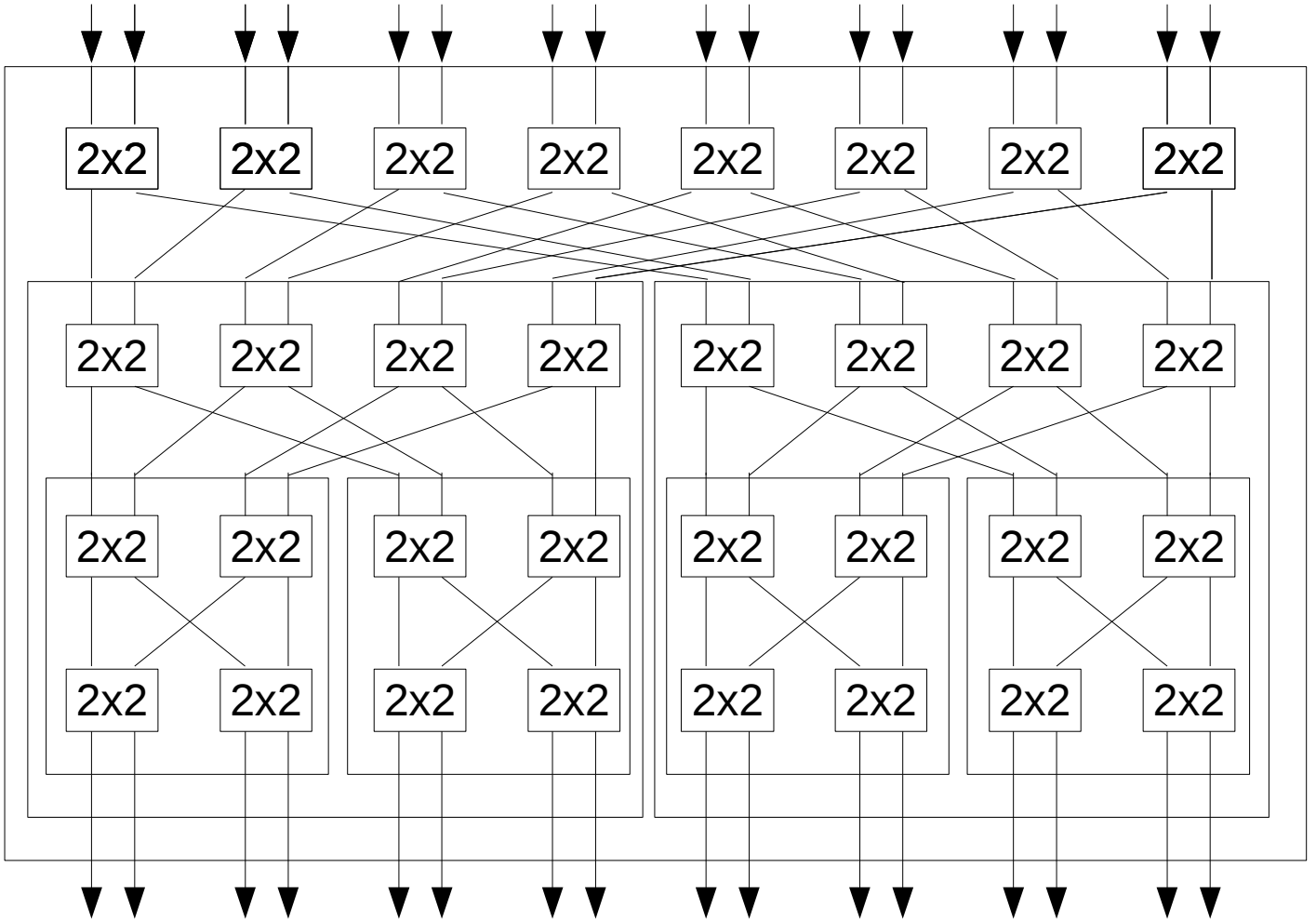
Switches with higher dimensions can be built using:
- **n n_way_dispatchers**
- **n n_way_mergers**

(see right picture)

[,,,] or instantiating a network of 2x2 switches (see next slide)



Switch implementation (2)

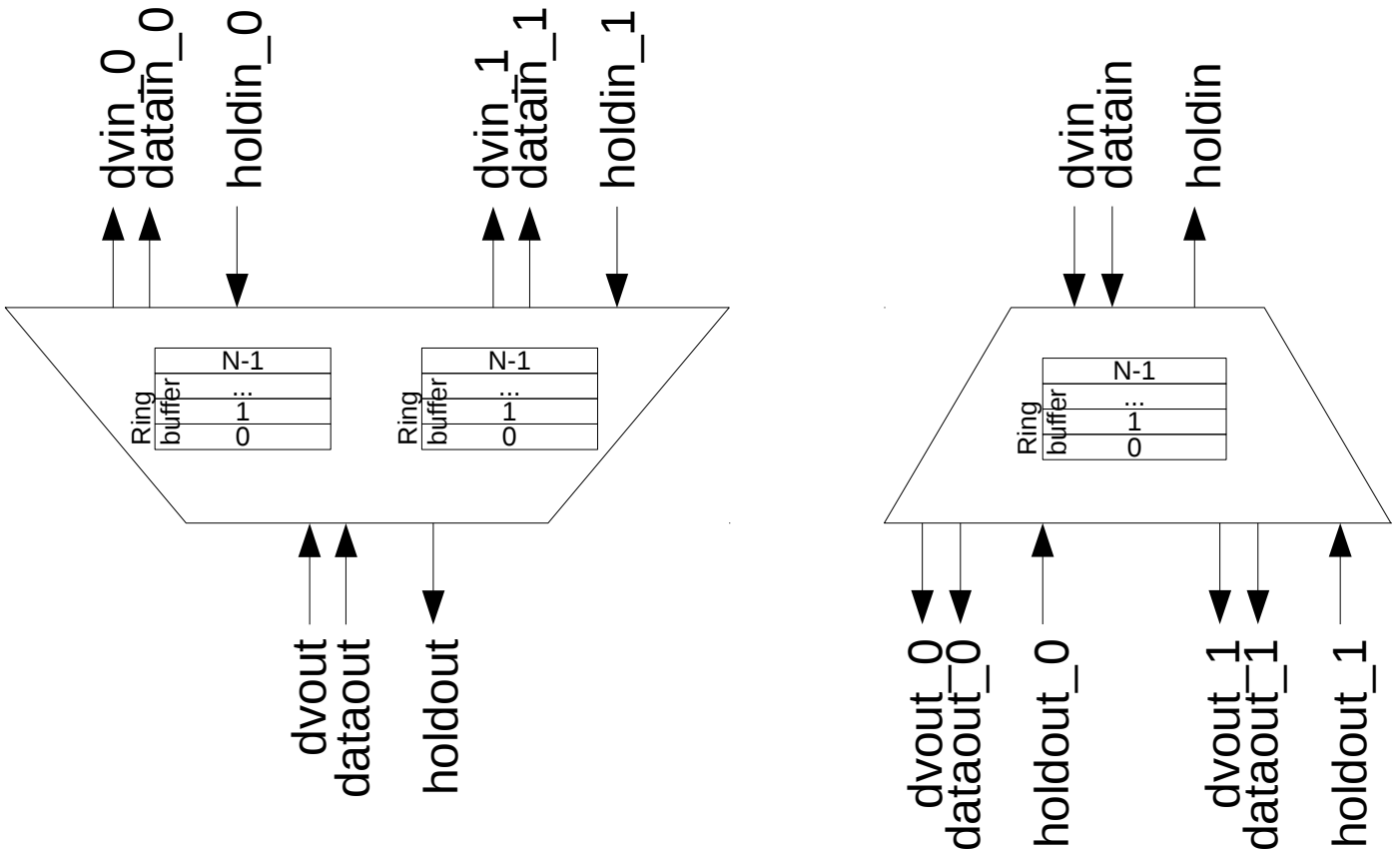


A generic $2^m x 2^m$ switch is built using:

- a layer of $\mathbf{2^{m-1}}$ **(2x2)_switches**
- a layer of **2** **$(2^{m-1}x2^{m-1})$_switches**

→ Total : $n/2 * \log_2(n)$ (2x2)_switches

Hold logic in the switch



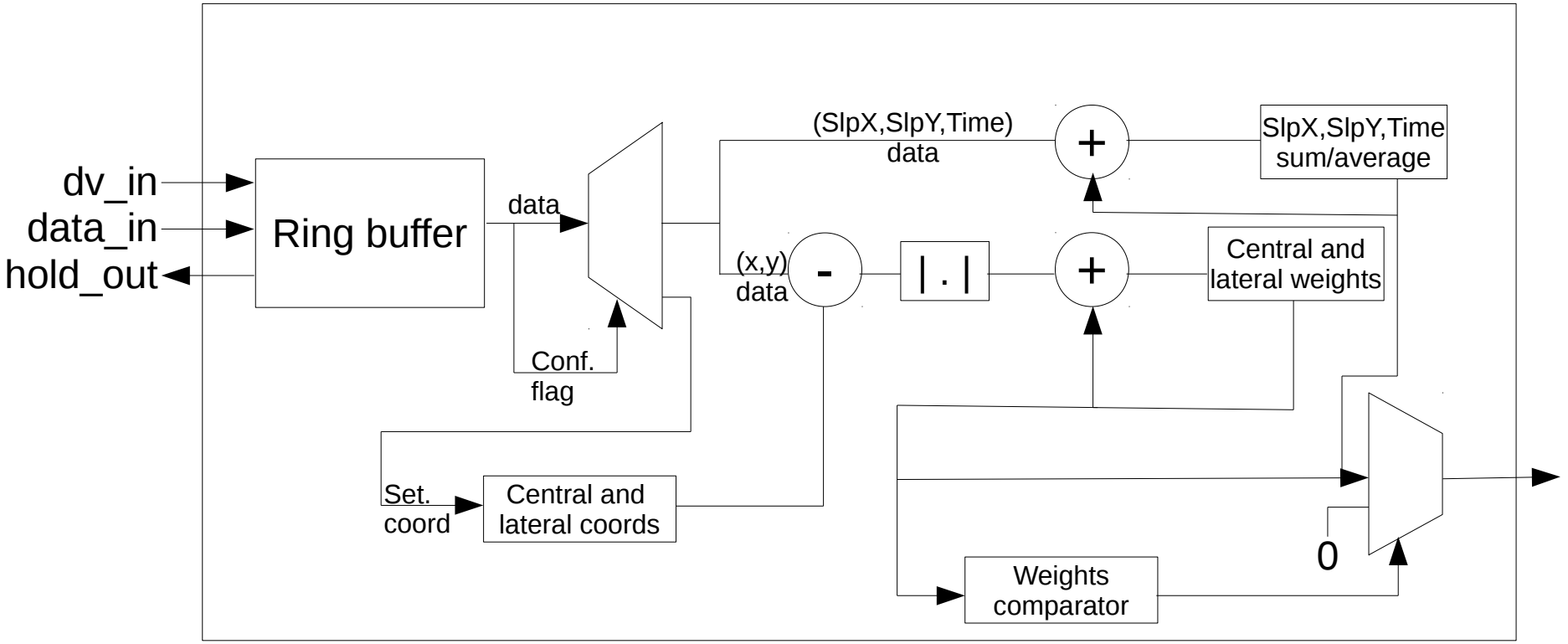
The **data flow through the switch** is managed by its basic components:

- Dispatchers and mergers have **a ring buffer for each input**
- If the **buffer is full** an **“hold signal”** is back propagated to the previous component

Engine scheme

The engine evaluates the **response of a cellular unit** and for laterl cells:

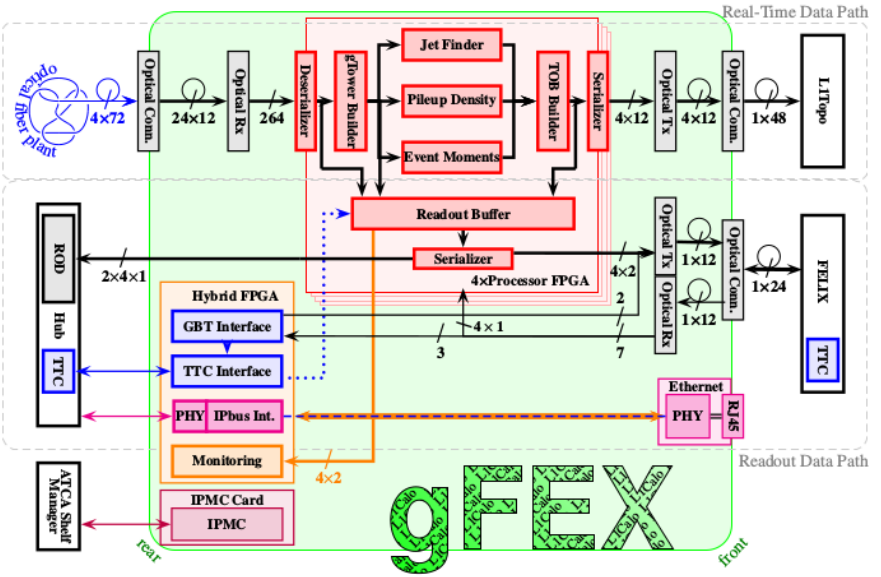
- The main apporach is described in the scheme
- Tracks information are retrieved by **interpolating the response and by averaging** the hit values (for SlpX, SlpY, Time)

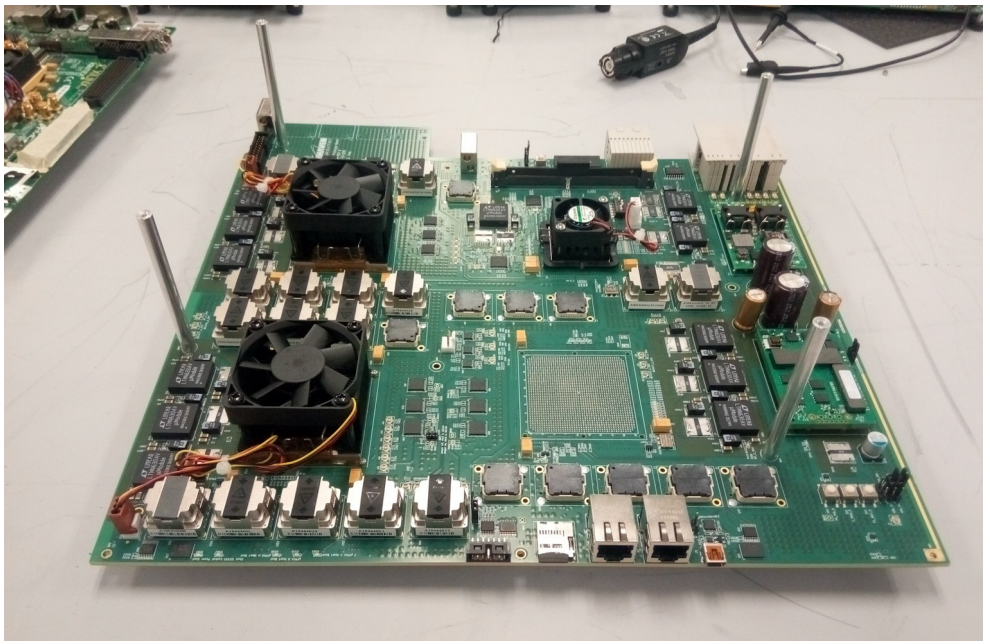


gFEX board

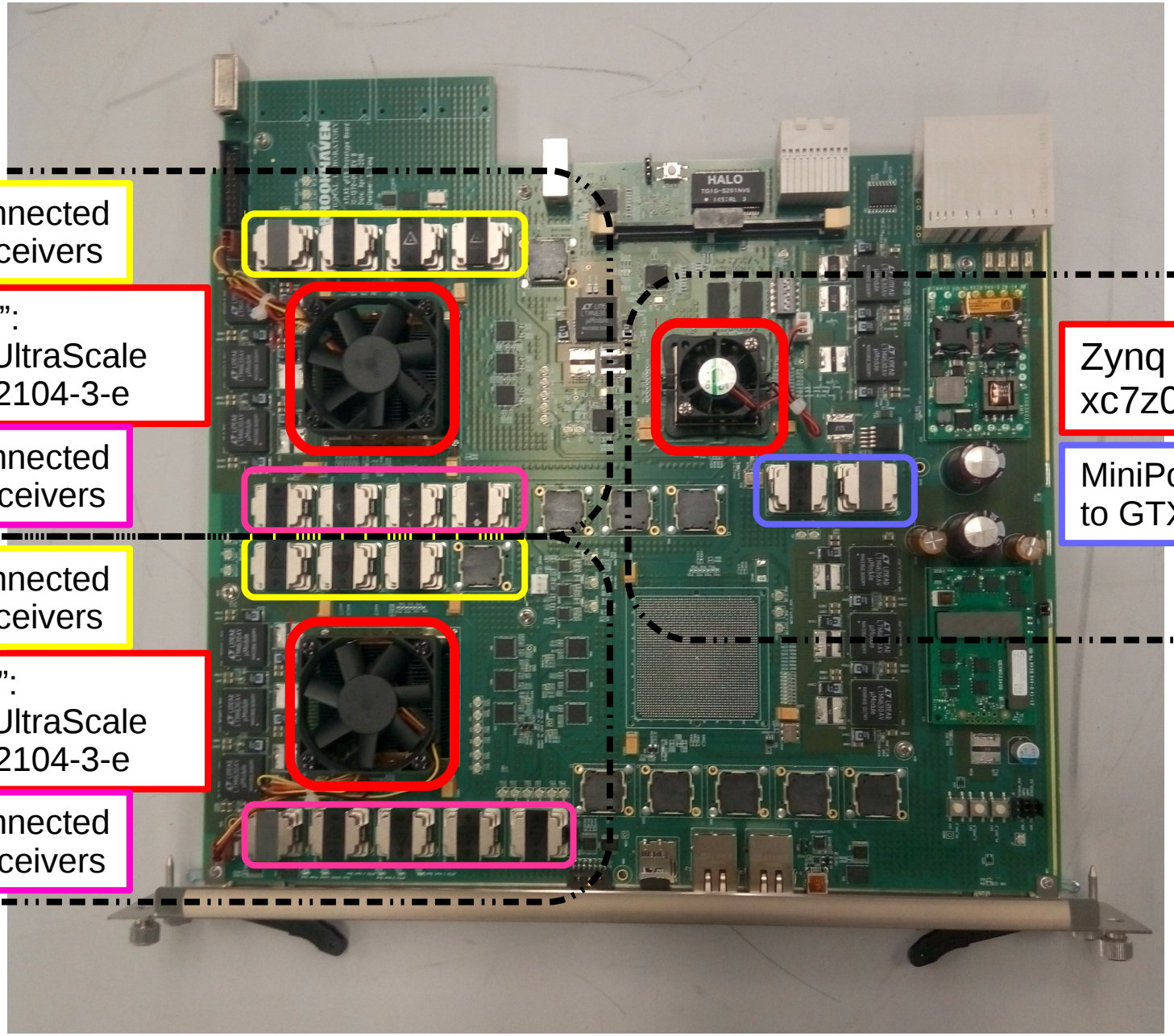
“global Feature eXtractor”:

- ATCA board, designed and produces at BNL for ATLAS Calorimeter Level 1 Trigger
- hosts 2 Virtex UltraScale and 1 Zynq FPGAs
- Up to ~2Tbps input data



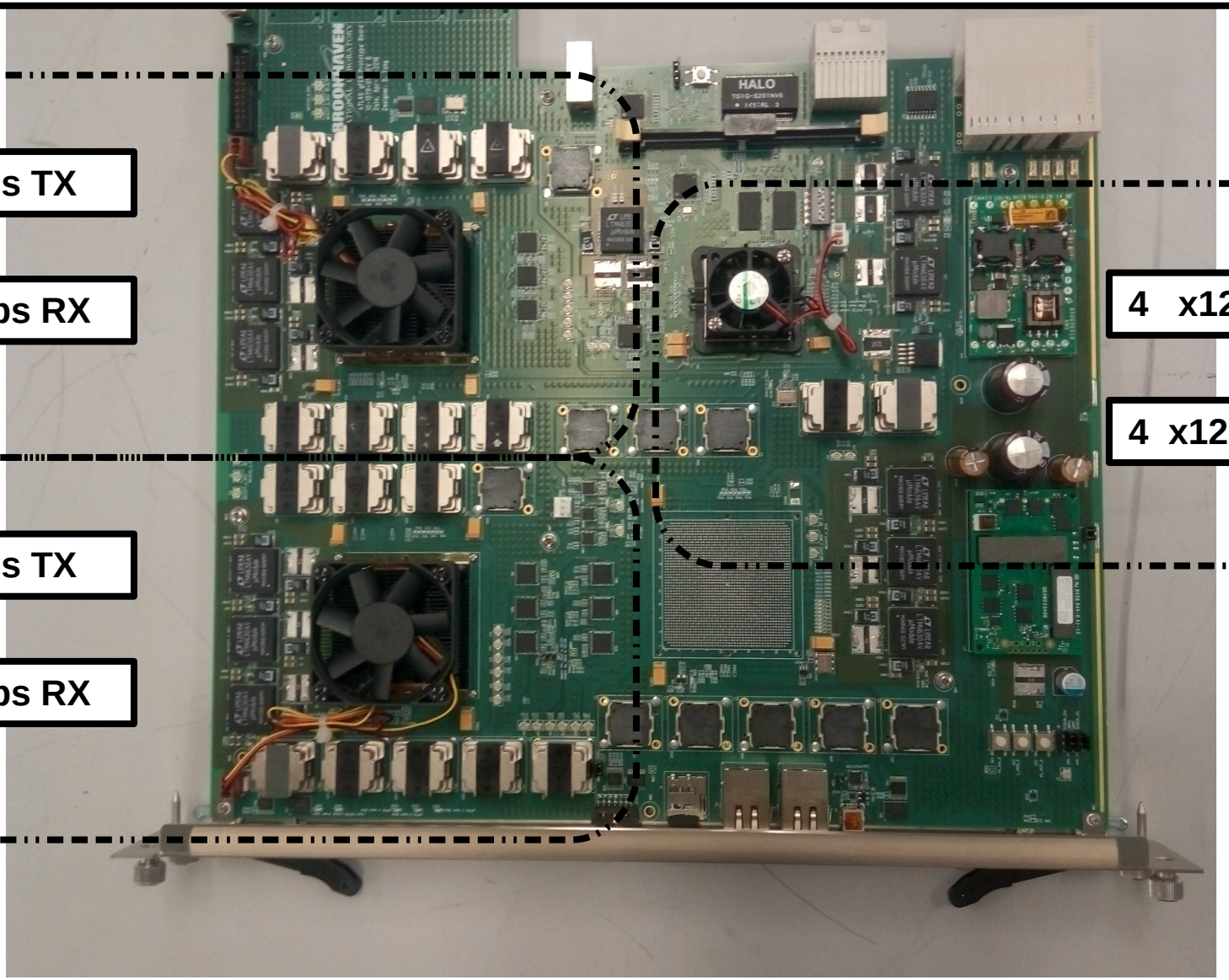


gFEX board overview



gFEX – external communication

MGT (GTX, GTH, GTY) transceivers connected to MiniPods for communication over optic fibers.



8 x12.8Gpbs TX

64 x12.8Gpbs RX

4 x12.8Gpbs TX

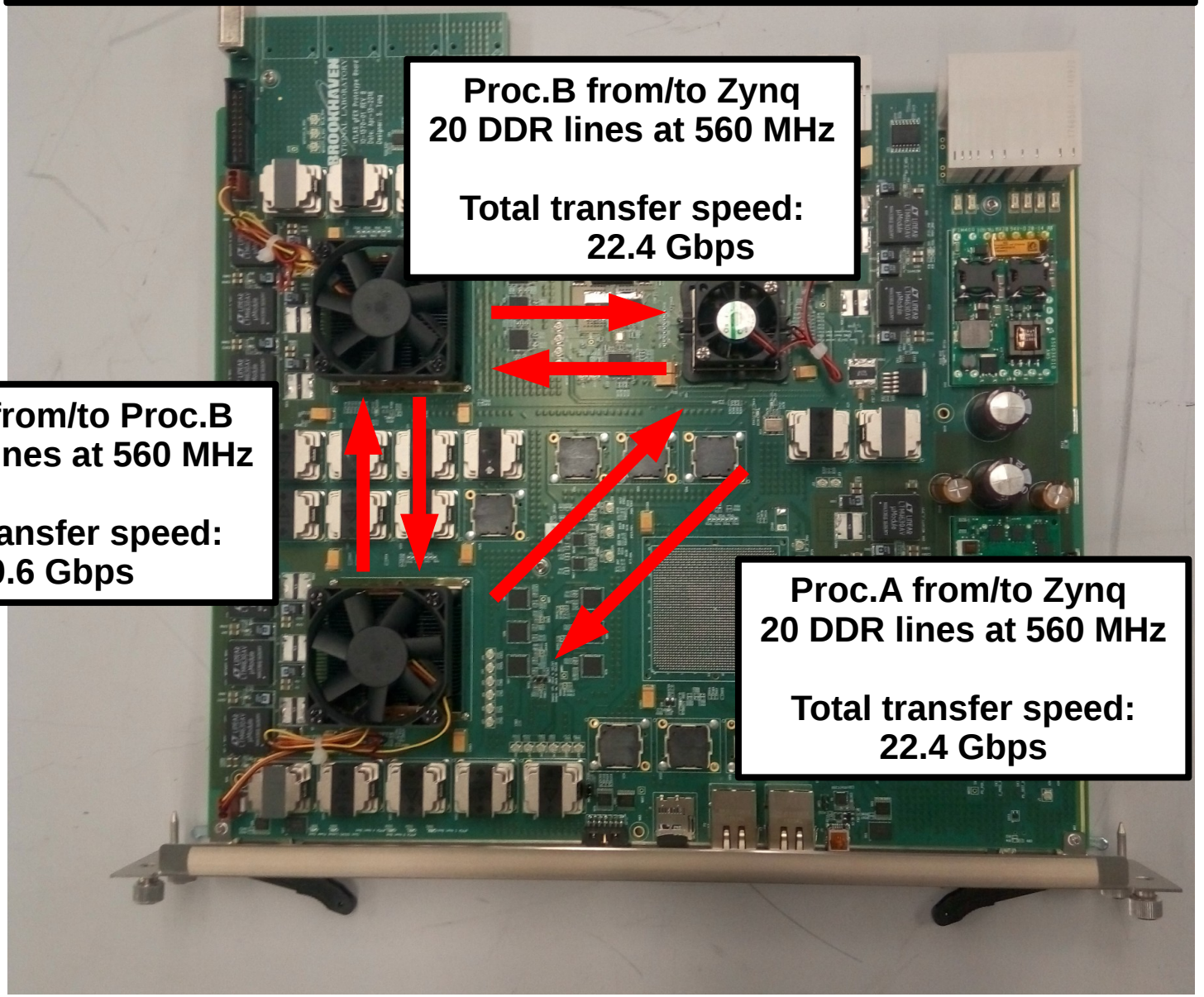
64 x12.8Gpbs RX

4 x12.8Gpbs TX

4 x12.8Gpbs RX

gFEX board – internal communication

Each DDR line can be used for RX or TX (but it is not bidirectional)



Proc.B from/to Zynq
20 DDR lines at 560 MHz

Total transfer speed:
22.4 Gbps

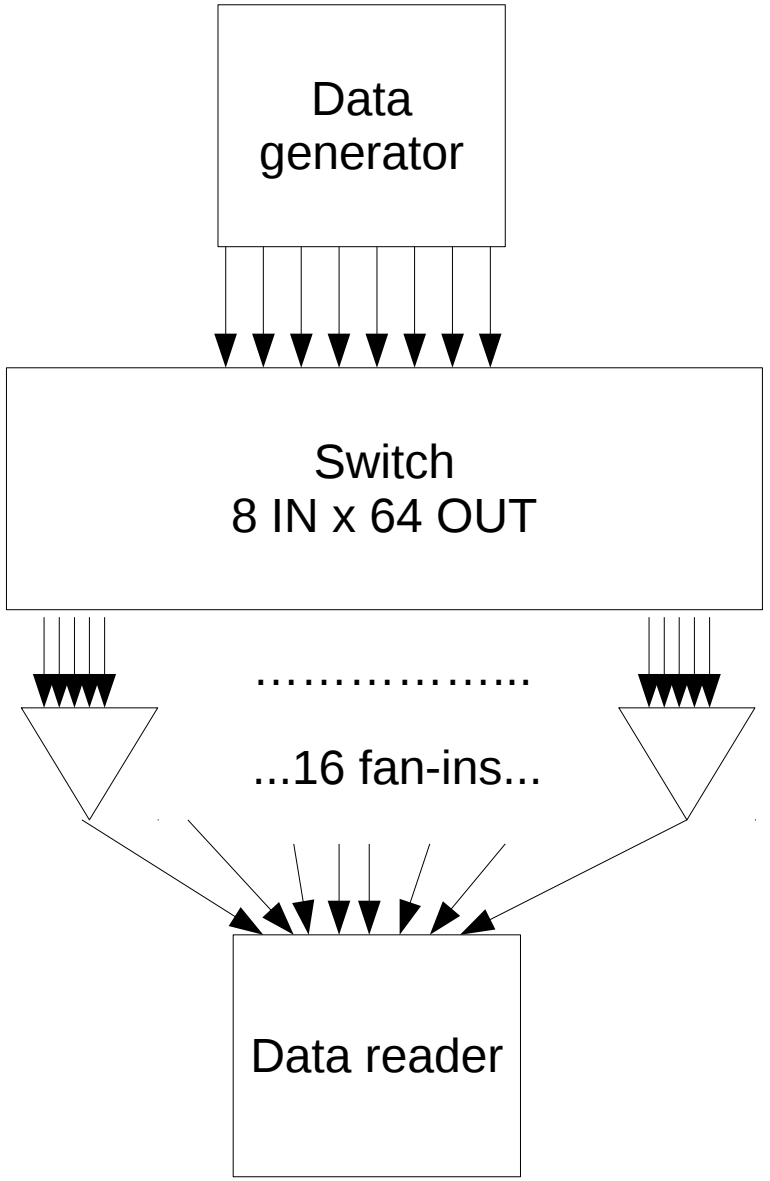
Proc.A from/to Proc.B
80 DDR lines at 560 MHz

Total transfer speed:
89.6 Gbps

Proc.A from/to Zynq
20 DDR lines at 560 MHz

Total transfer speed:
22.4 Gbps

Switch test on gFEX board



Architecture overview:

- Data generator (8 outputs)
- Switch “8 inputs x 64 outputs”
- 16 “4 inputs” fan-ins
- Data reader (16 inputs)

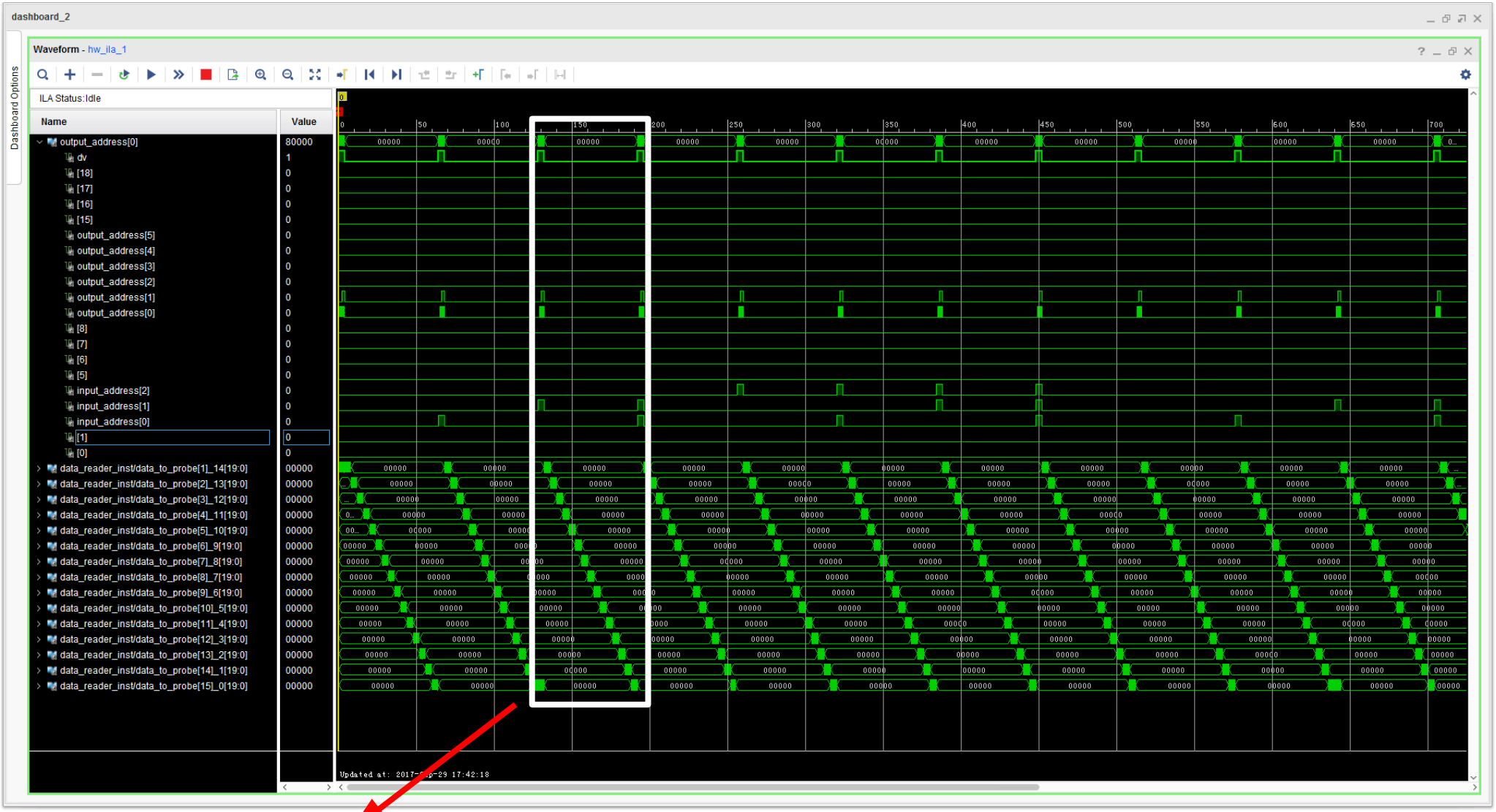
All the modules have been implemented in the same FPGA (Processor A)

Each input of the Data reader is read out from the PC through an ILA core (Integrated Logic Analyzer)

Simple test performed:

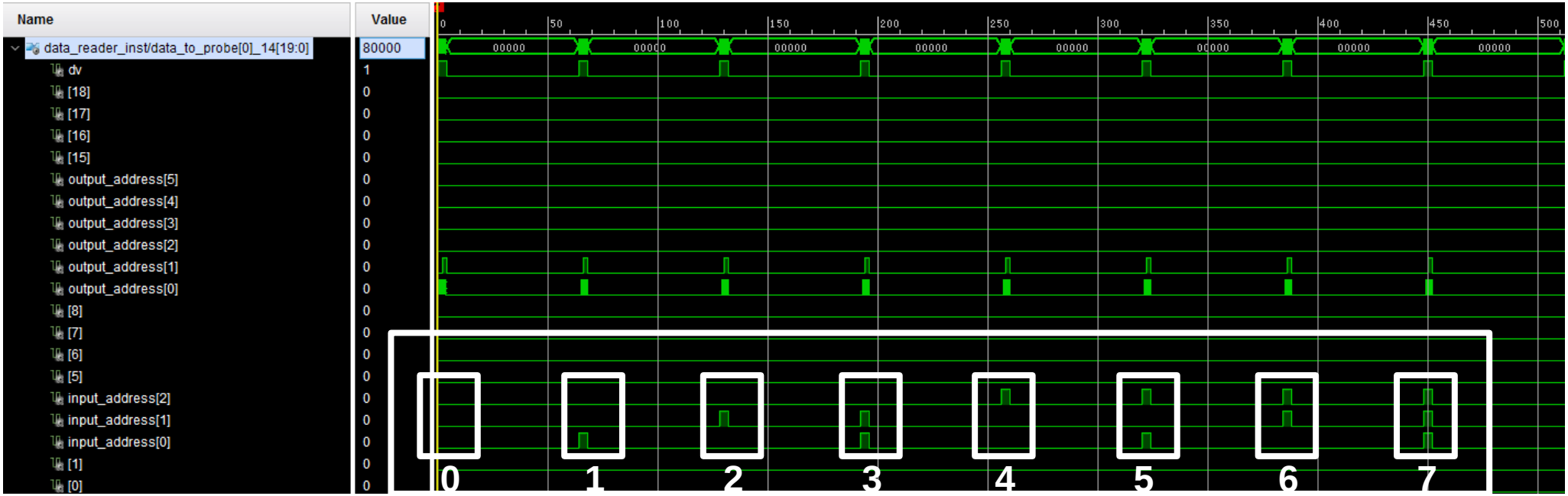
- One data generated at each clock cycle
- Loop over the switch inputs
- For each input data with all the possible “addresses” are generated (the address determines the data path in the switch)

Switch test results



Data from same input and with different addresses reach the proper outputs

Switch test results - detail



All the data from different inputs and with same address reach the same outptut

The switch has been tested running at 480 MHz clock speed and proved to work.

The firmware has been also tested with 560 MHz clock and data were corrupted or didn't reach the proper output

Future plans for implementation on gFEX

What has been done:
- Communication via MGT transceivers implemented.
- Communication via parallel data bus implemented.
- Small switch module implemented and tested with data generated and read inside the same FPGA.

What is missing:
- gFEX board, in transit from BNL to Milano
- Stub construction as part of the artificial retina or from a previous stage of processing (as part of the DAQ)

Future plans and tests:
- Implementation of the full Artificial Retina using multiple FPGAs on the gFEX board:
  Example 1: switch in one FPGA, engines in the other.
  Example 2: distributed switch and engine resources in both FPGAs
- Low-level simulation of the response with data from the LHCb VELO Upgrade detector
  4D/3D simulations with/without the time information of the hits
- Test/evaluation of the maximum data/event rate that can be handled by the system (up to the expected 40MHz LHC bunch crossing rate)

Conclusion and future plans

The Artificial Retina algorithm has been implemented in FPGA and tested for a 2D tracking system at SPS with good results.

The Artificial Retina can be applied for 3D/4D tracking system for real-time track reconstruction:

- The use of the stubs is introduced to help in the pattern recognition
- The introduction of the time information increases the purity of the reconstruction and allows the measurement of the track time (to be used in the vertex reconstruction)

The firmware implementation is almost complete:

- On-board test of the switch performed
- Test of the maximum exploitable event rate with data from external board to be performed