ARCADIA FNAL meeting

05-12-24

S. Ciarlantini, C. Pantouvakis, M. Rignanese, A. Zingaretti, D. Chiappara, C. Bonini, P. Giubilato, S. Mattiazzo, J. Wyss, D. Pantano, P. Azzi, R. Turrisi
INFN Sezione di Padova

ARCADIA

Istituto Nazionale di Fisica Nucleare

TB analysis: status

<u>What's new</u>

- Study on long clusters: extended timestamp correction
- Analysis on FEB Parameter scan data
  - IBIAS and IFB
  - ID

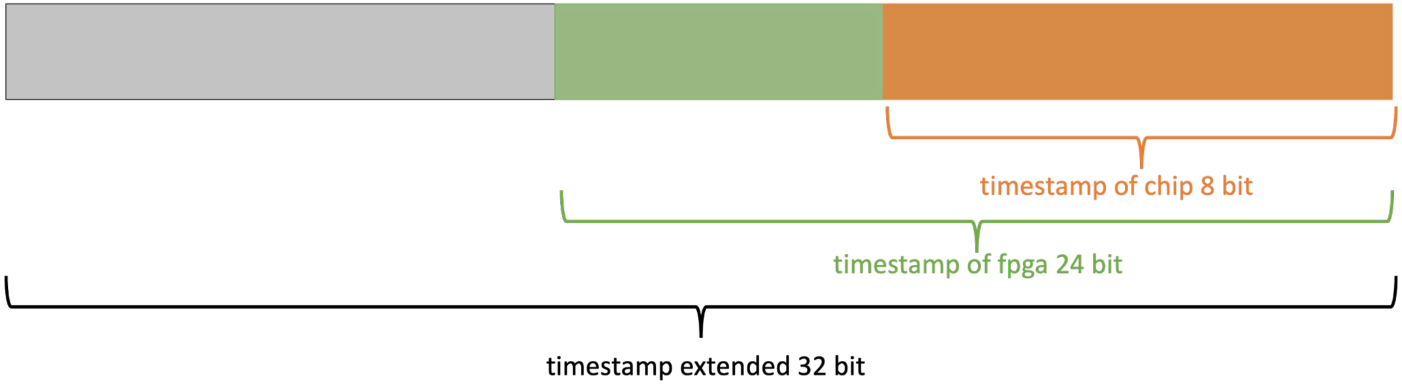
Long clusters:
extended timestamp correction

Recap on ARCADIA timestamps: single chip DAQ

The flow of timestamp association is the following

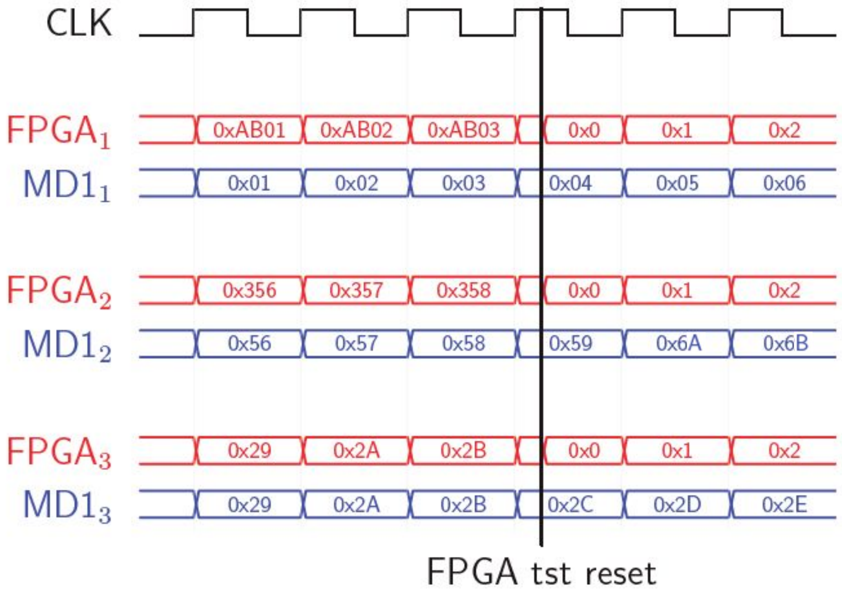
1. *Hit arrives on the pixel and information is sent to the periphery*
   The periphery attaches the **ts_chip** → 8 bit long
2. *Hit information is transmitted to the FPGA*
   The FPGA attaches its timestamp **(ts_fpga)** → 24 bit long
3. *Analysis online software decodes data*
   It extends the timestamp and creates **ts_ext** using ts_fpga and considering how many times ts_fpga has done a rollover → 32 bit long

During the analysis we use ts_ext as timestamp



Recap on ARCADIA timestamps: telescope (three chip) DAQ

To synchronize the three chips for telescope DAQ system, the FPGA timestamp is reset for all chips at the same time, then, to keep the synchronization, the timestamp of the chip is ignored and only the FPGA timestamp is considered when computing the ts_ext.



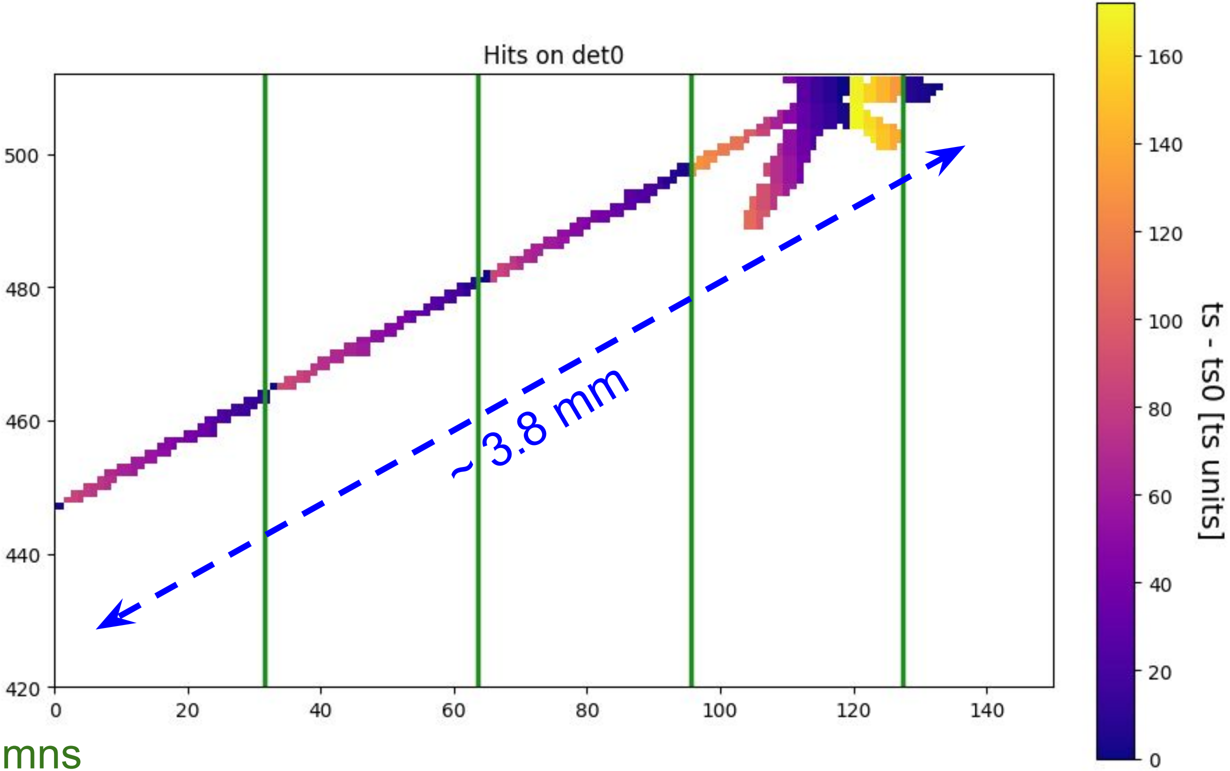
To keep synchronization, in the current software the extended timestamp is based on the FPGA timestamp, neglecting completely the chip timestamp.

Issue:

When there are too many hits coming from the same sector, there is too much "traffic" in the periphery and therefore there is a significant delay between the real timestamp of the hit and the FPGA timestamp that is assigned to the packets.

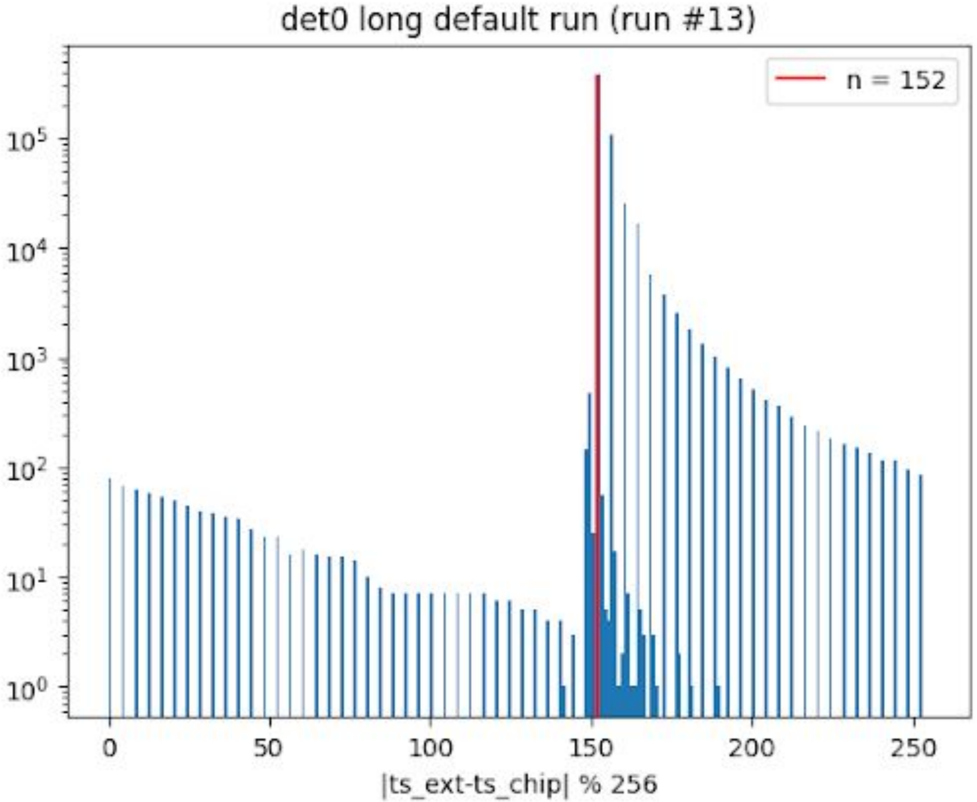
Long cluster event on detector 0

Hits for high multiplicity event on detector 0 are very spread in time.
As we go to larger timestamps (up to 170 timestamp units, **about 34 us**), the complete tracks on det0 are recovered.
With our time window of 41 for clustering, we cut this single event in many clusters.



sector = 32 columns

Histogram of |*ts_ext - ts_chip*| *% 256* for all packets of detector 0 coming from the same single run.
|*ts_ext - ts_chip*| *% 256* checks the last 8 bits of the difference between the ts_ext and ts_chip: if <u>no delay</u> occurs with timestamps, this <u>difference is constant</u>.
All the entries that are not in the peak (n = 152) are packets that were in queue.

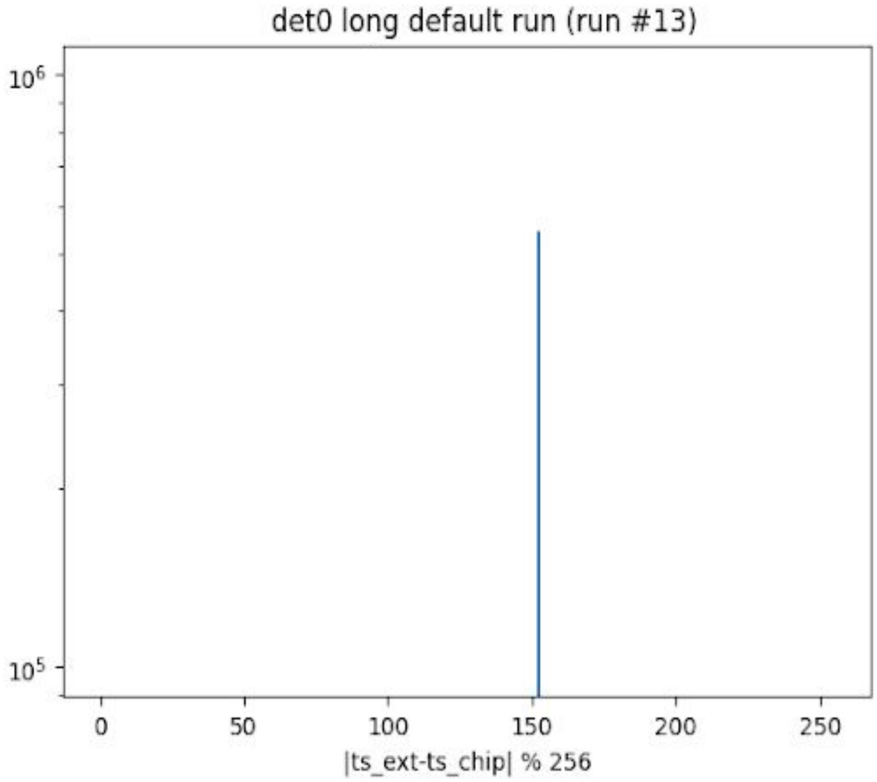


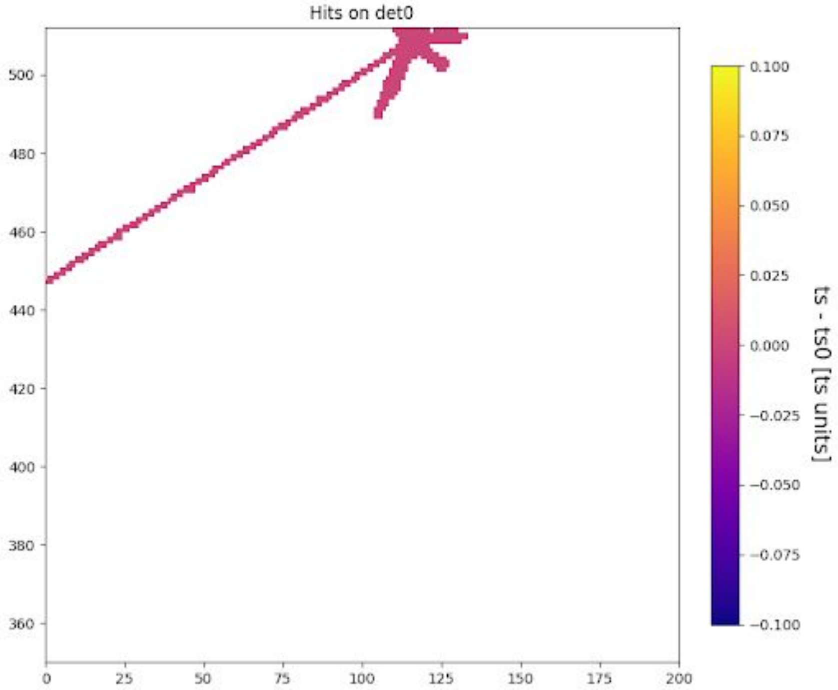
It is possible to correct for this issue, taking into account the offset between FPGA and chip timestamps

1) find the peak **n** of the histogram
2) compute **d = |ts_ext-ts_chip|%256 - n**
3) apply the shift
   a) d > 0 → ts_ext - d
   b) d < 0 → ts_ext - (256 +d)

This correction should be implemented in the online DAQ software

After the offline correction



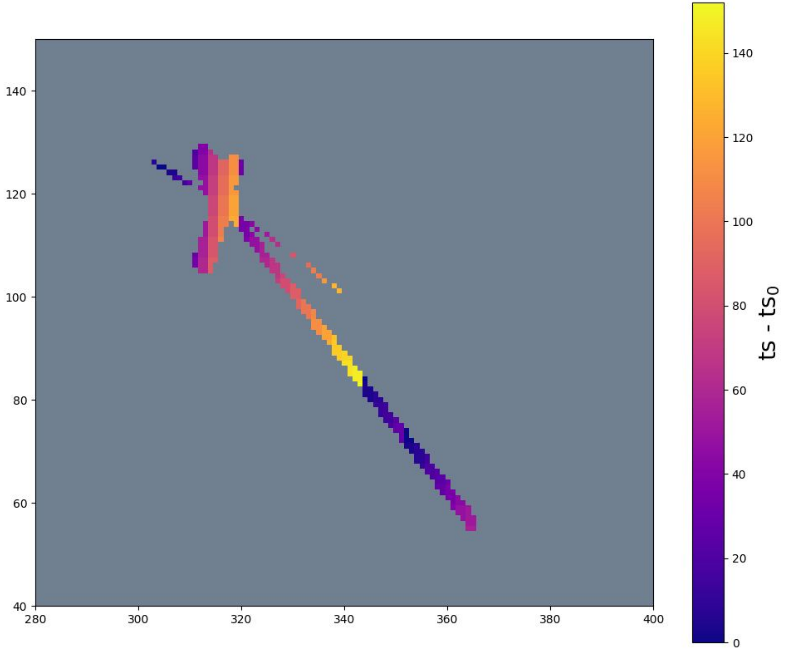


After the shift, all entries in the peak bin (n=152): there is no more delay between ts_chip and ts_ext as should be.

Single cluster with all hits recorded at the same timestamp!
OK!

Another example

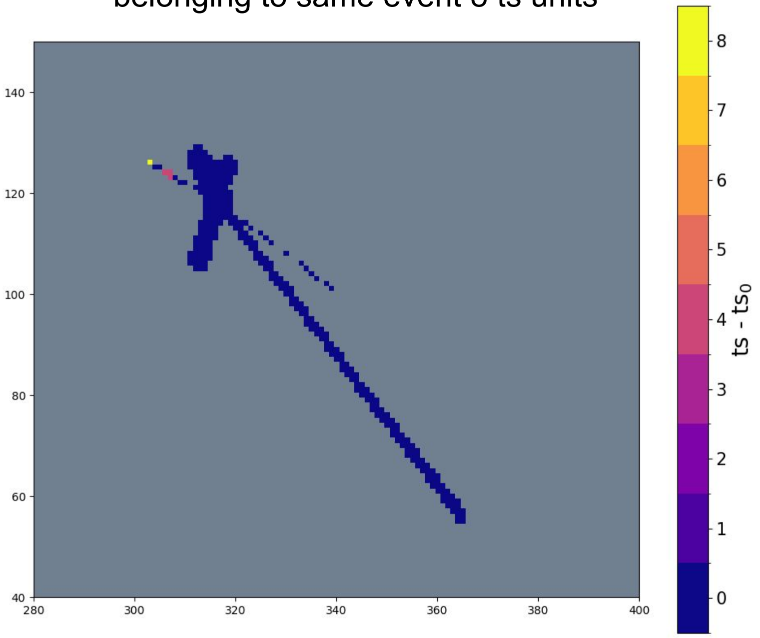
Before extended
timestamp correction



After clusterization, 17 clusters

After extended
timestamp correction

maximum difference between hits
belonging to same event 8 ts units



After clusterization, 6 clusters
max multiplicity 308

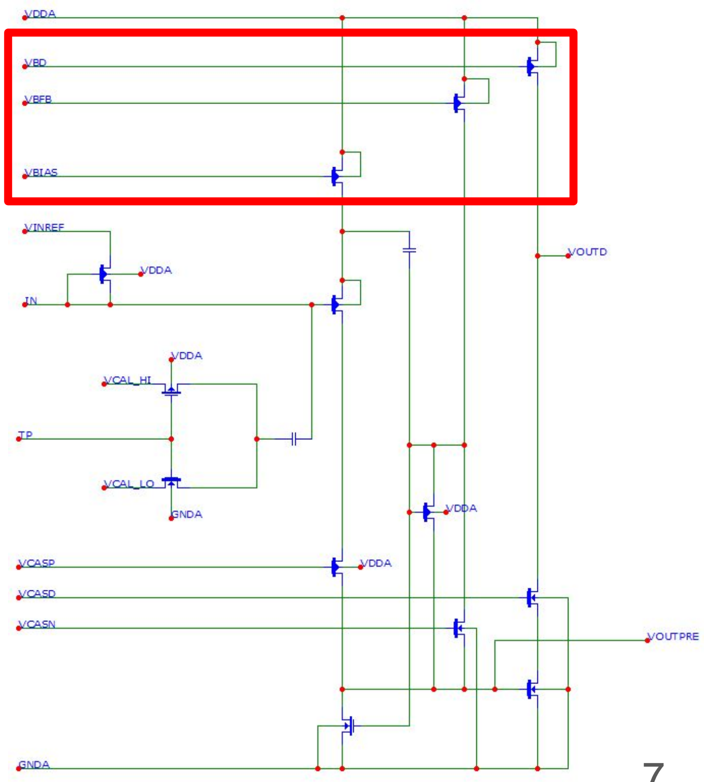
FEB Parameters scan on DUT

FEB Parameters

Relevant parameters for pixel preamplifier gain:

- **IBIAS/IFB** are respectively the main currents in the preamplifier and the bias current in the feedback branch. IBIAS has a strong impact on the preamplifier dead time.
  IBIAS and IFB have to be set to the same value.
- **ID** controls the current in the discriminator.
  A higher value reduces the slew rate at discriminator output.

Gain increases when
- ❖ IBIAS/IFB decreases
- ❖ ID decreases

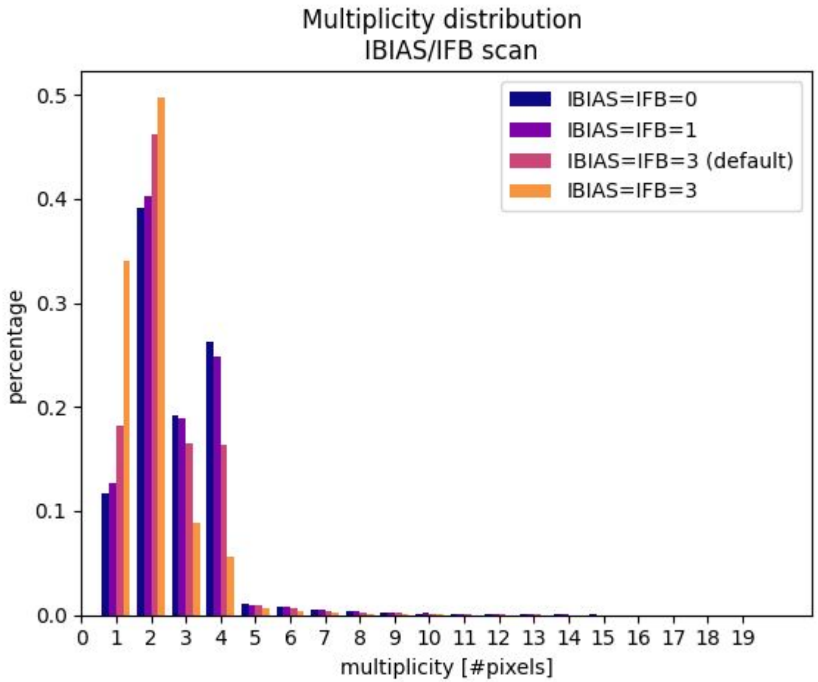


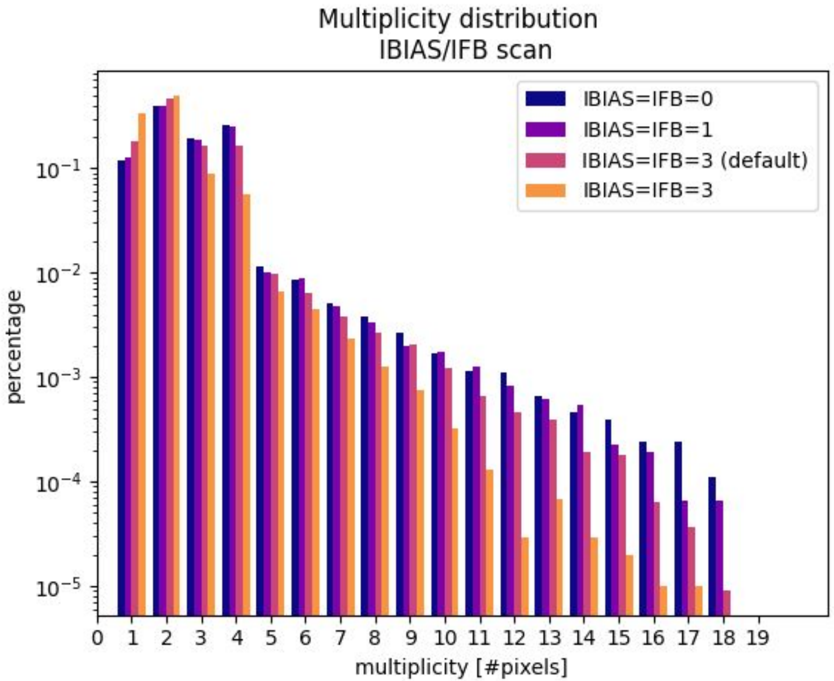
Measurements at test beam:

Two parameters were scanned on DUT

- IBIAS/IFB → scan of values = 0, 1, 2 (default), 3
- ID → scan of values = 0 (default), 1, 2, 3

FEB parameters: **IBIAS and IFB** scan
DUT Clusters (associated to tracks) distributions



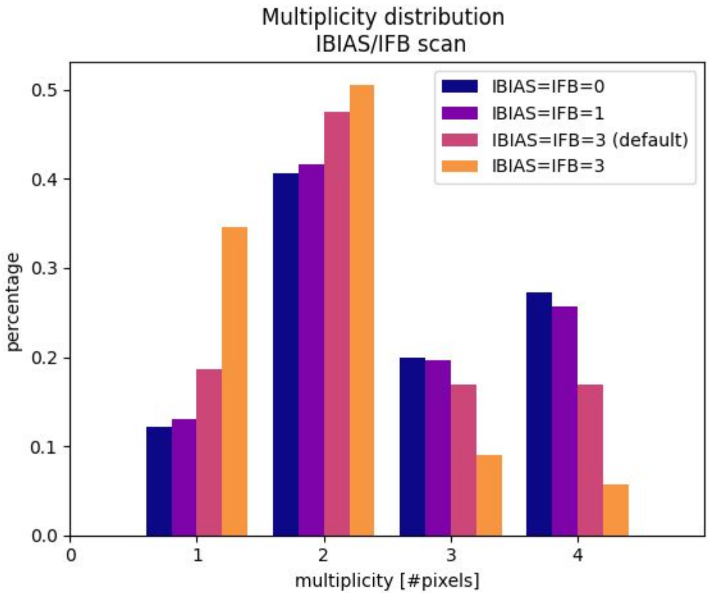


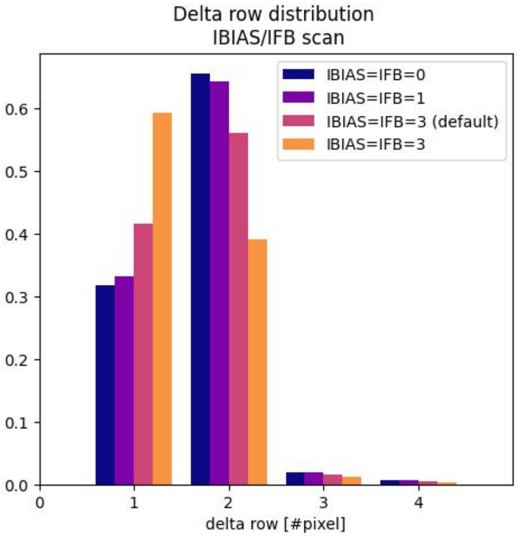
Average mult per IBIAS=IFB=0: 2.797 +/- 0.004
Average mult per IBIAS=IFB=1: 2.740 +/- 0.004
Average mult per IBIAS=IFB=2: 2.447 +/- 0.004
Average mult per IBIAS=IFB=3: 1.929 +/- 0.003

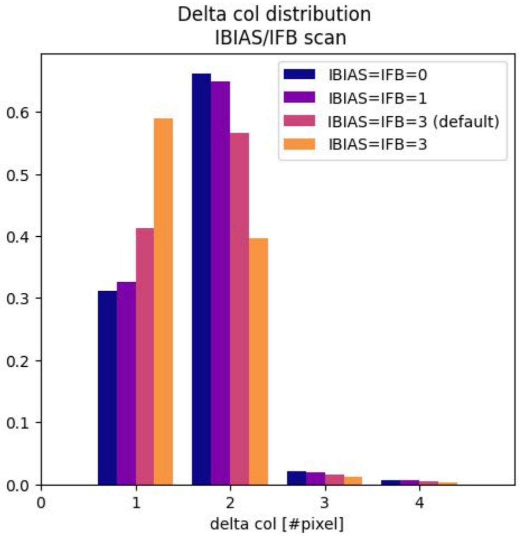
↑ gain
pixel amplifier

Avg. multiplicity increases as pixel gain increases ✅

FEB parameters: **IBIAS and IFB** scan
DUT Clusters multiplicity <= 4 distribution





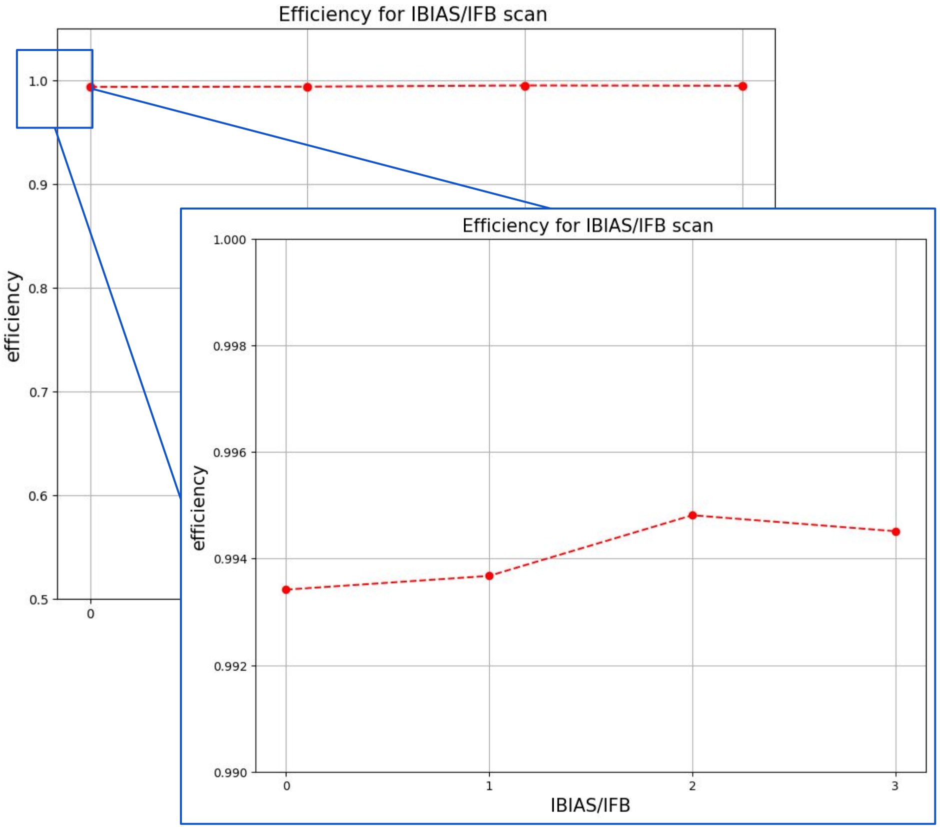


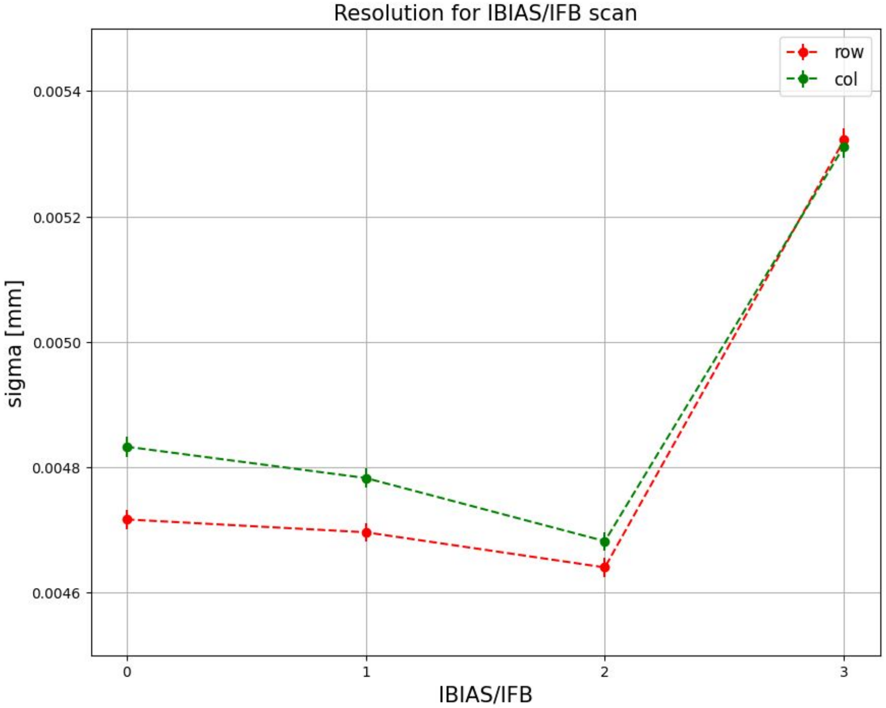
Clusters with mult <= 4 for IBIAS = IFB = 0: 96.2 %
Clusters with mult <= 4 for IBIAS = IFB = 1: 96.5 %
Clusters with mult <= 4 for IBIAS = IFB = 2: 97.2 %
Clusters with mult <= 4 for IBIAS = IFB = 3: 98.4 %

gain
pixel amplifier

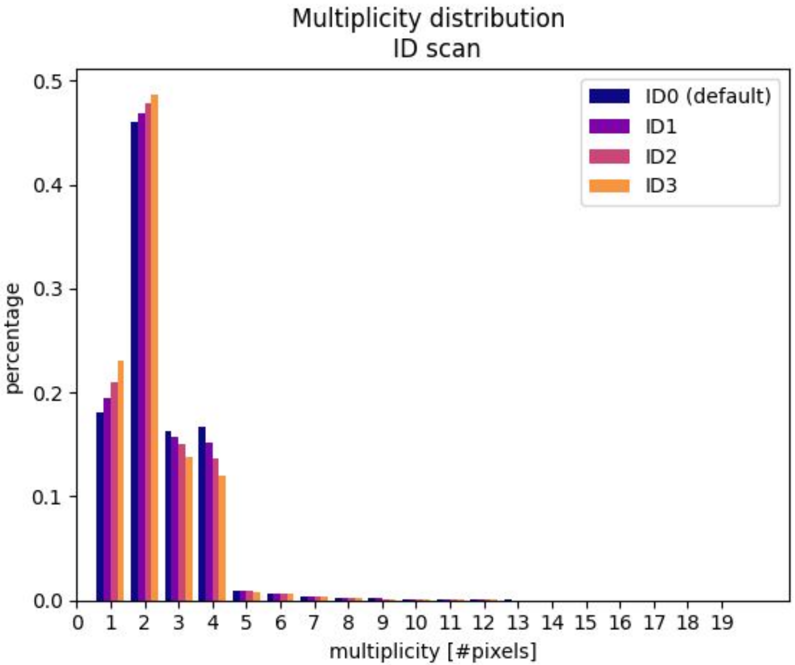
IBIAS=IFB=3 (low gain) shows more single pixel clusters respect to 0 (high gain) ✅

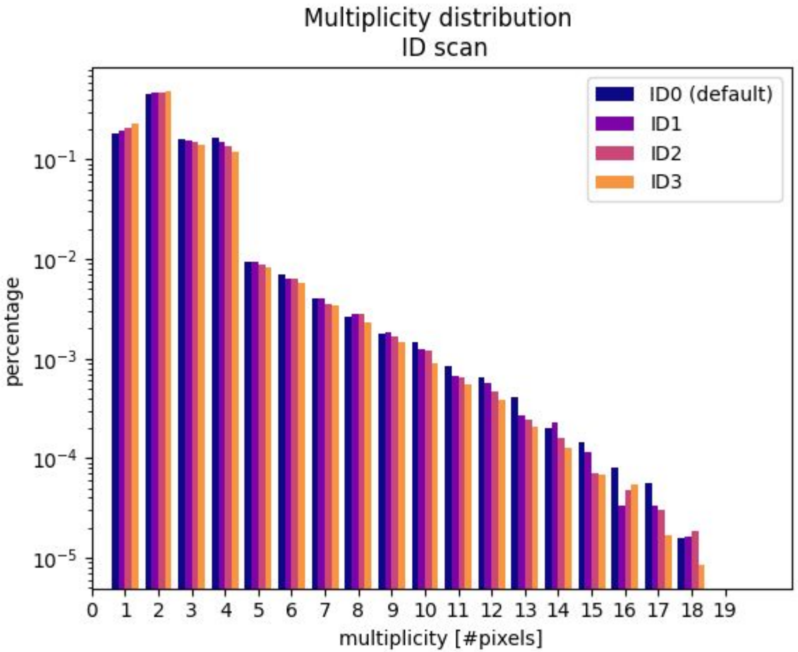
FEB parameters: **IBIAS and IFB** scan
Efficiency and resolution





FEB parameters: ID scan
DUT Clusters (associated to tracks) distributions



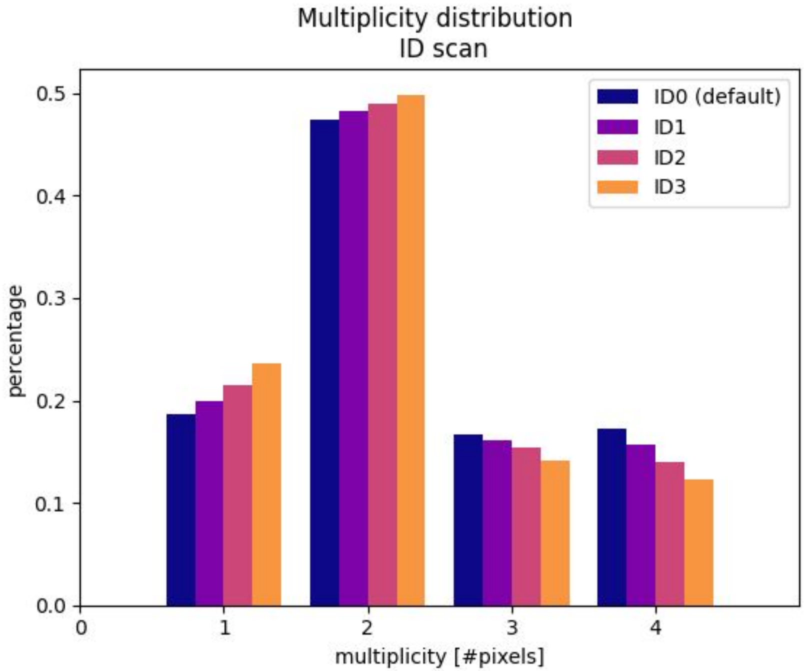


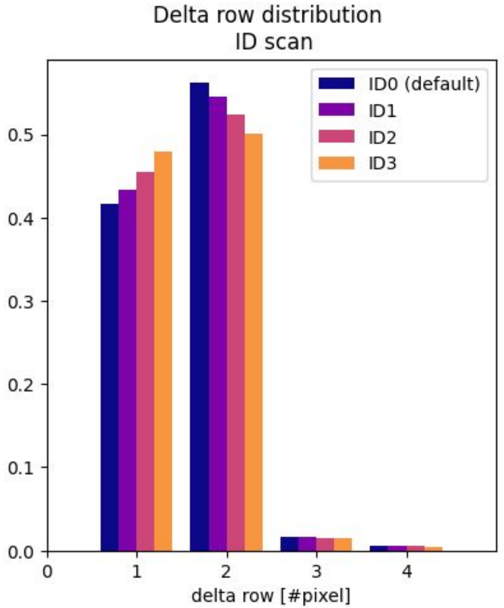
Average mult per ID = 0: 2.457 +/- 0.004
Average mult per ID = 1: 2.401 +/- 0.004
Average mult per ID = 2: 2.338 +/- 0.002
Average mult per ID = 3: 2.259 +/- 0.002

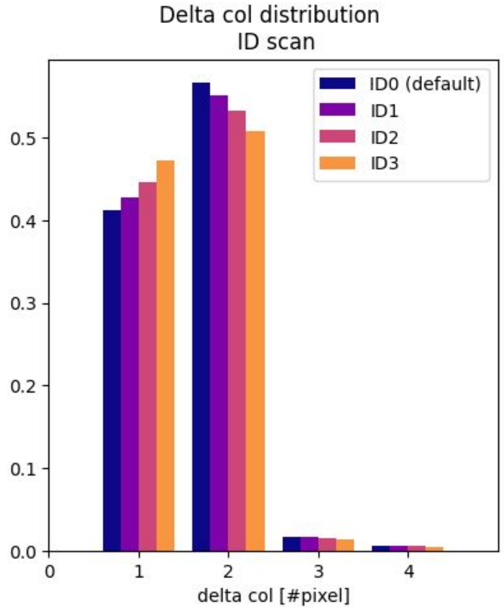
gain
pixel amplifier

Avg. multiplicity increases as pixel gain increases

FEB parameters: ID scan
DUT Clusters multiplicity <= 4 distribution







Percentage of clusters with mult <= 4 for ID0: 0.971
Percentage of clusters with mult <= 4 for ID1: 0.972
Percentage of clusters with mult <= 4 for ID2: 0.974
Percentage of clusters with mult <= 4 for ID3: 0.976

gain
pixel amplifier

ID=3 (low gain) shows more single pixel clusters respect to 0 (high gain)

FEB parameters: ID scan
Efficiency and resolution