

Faulty core columns in CROC quad assemblies

B. Raciti, IT ASIC & Electronics System – 8th June 2023

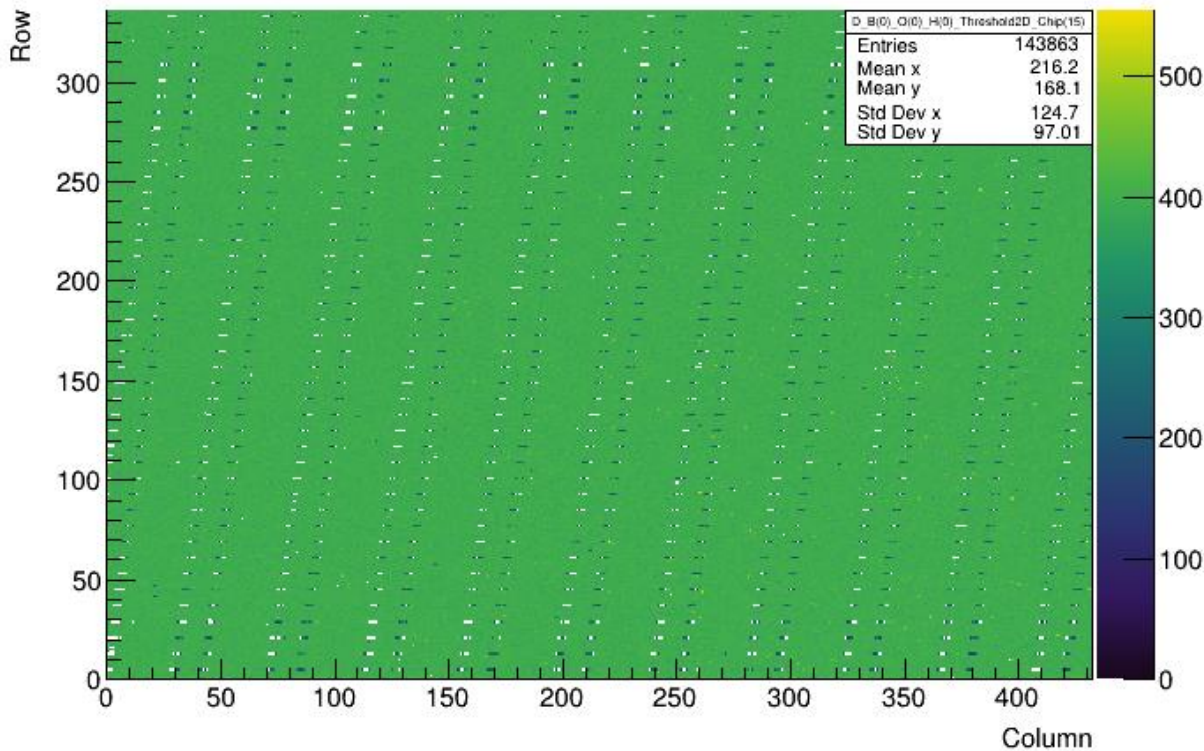
Injection pattern and faulty core columns

Issue:

- All quad assemblies tested so far presented an odd injection pattern independent of the sensor design and HDI (LF - HPK, TEPX - TBPX)
→ chip related issue
- This is due to the presence of faulty pixels in some of the core columns sending corrupted data

A quick test can be done by running a **pixelalive** scan activating the **flag for the data integrity**.

D_B(0)_O(0)_H(0)_Threshold Map_Chip(15)



Each digit corresponds to a core column (8 cols)

```

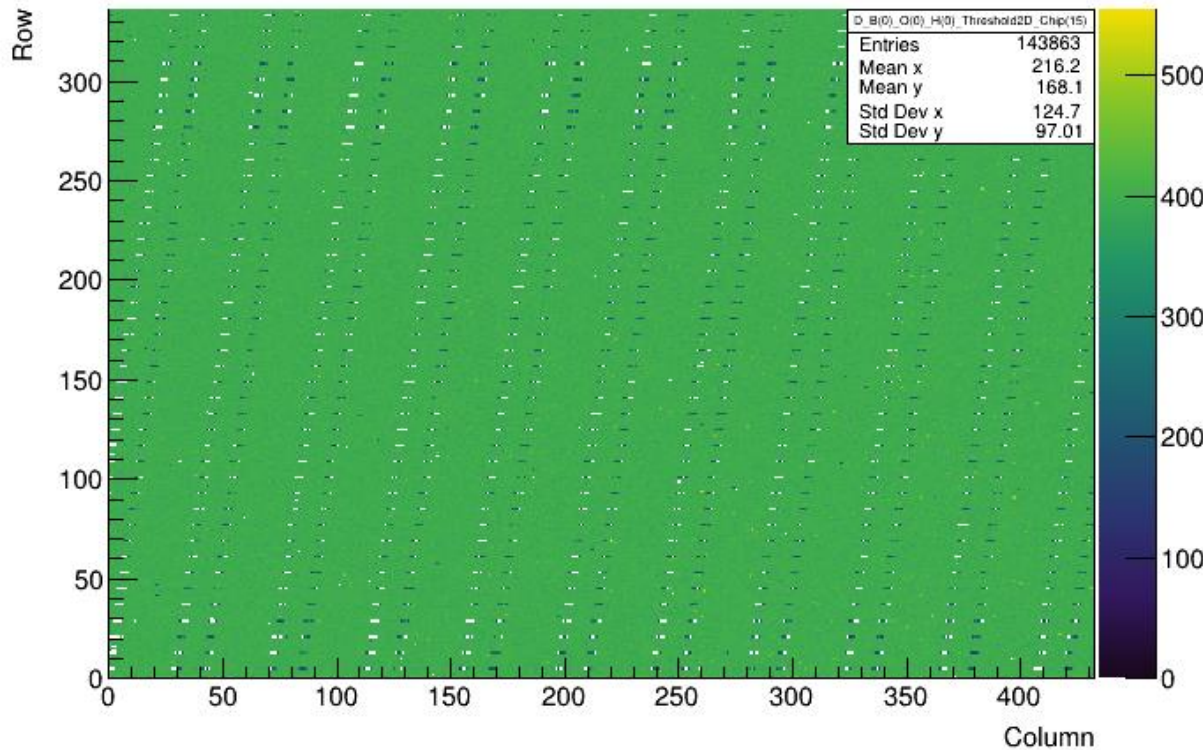
00:03:55 | E | No data collected --> retry
00:03:55 | E | No data collected --> retry
00:03:55 | E | No data collected --> retry
00:03:55 | E | No data collected --> retry
00:03:55 | E | No data collected --> retry
00:03:55 | E | No data collected --> retry
00:03:55 | E | No data collected --> retry
00:03:55 | E | No data collected --> retry
00:03:55 | E | No data collected --> retry
00:03:55 | E | No data collected --> retry
00:03:55 | E | --> Reached maximum number of attempts (10) without success
00:03:55 | I | Results for [board/opticalGroup/hybrid/chip 0/0/0/15]
00:03:55 | I | --> EN_CORE_COL_0 value = 1111111111111110 (0 = disabled)
00:03:55 | I | --> EN_CORE_COL_1 value = 1111111111111111 (0 = disabled)
00:03:55 | I | --> EN_CORE_COL_2 value = 1111111111111111 (0 = disabled)
00:03:55 | I | --> EN_CORE_COL_3 value = 111111 (0 = disabled)
00:03:55 | I | --> Done
  
```

Injection pattern and faulty core columns

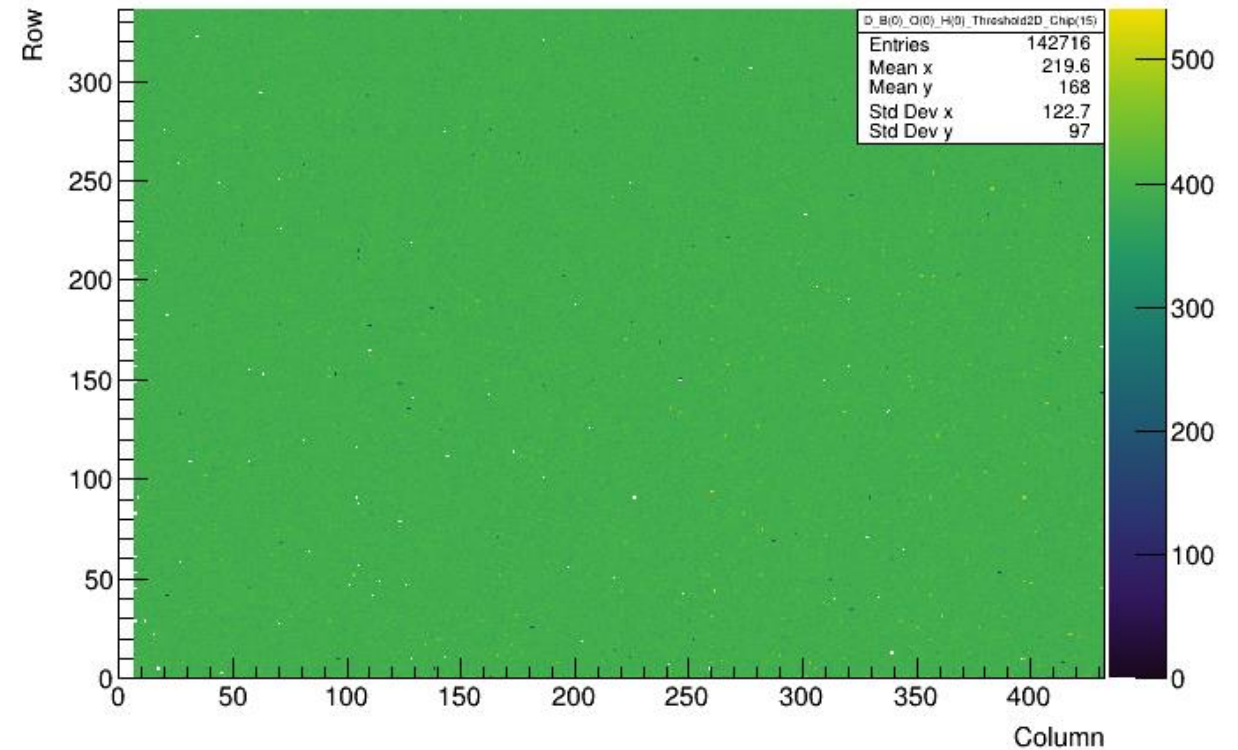
Just a handful of pixels are faulty (otherwise none of the pixels in these core columns would be tunable).

- These pixels are neither noisy nor stuck
 - they **send corrupted data** that cannot be decoded
- Since multiple pixels are injected simultaneously, the entire batch of data has to be **discarded**
 - origin of the injection pattern and complete failure of the noise scan (simultaneous readout of the entire matrix)

D_B(0)_O(0)_H(0)_Threshold Map_Chip(15)



D_B(0)_O(0)_H(0)_Threshold Map_Chip(15)

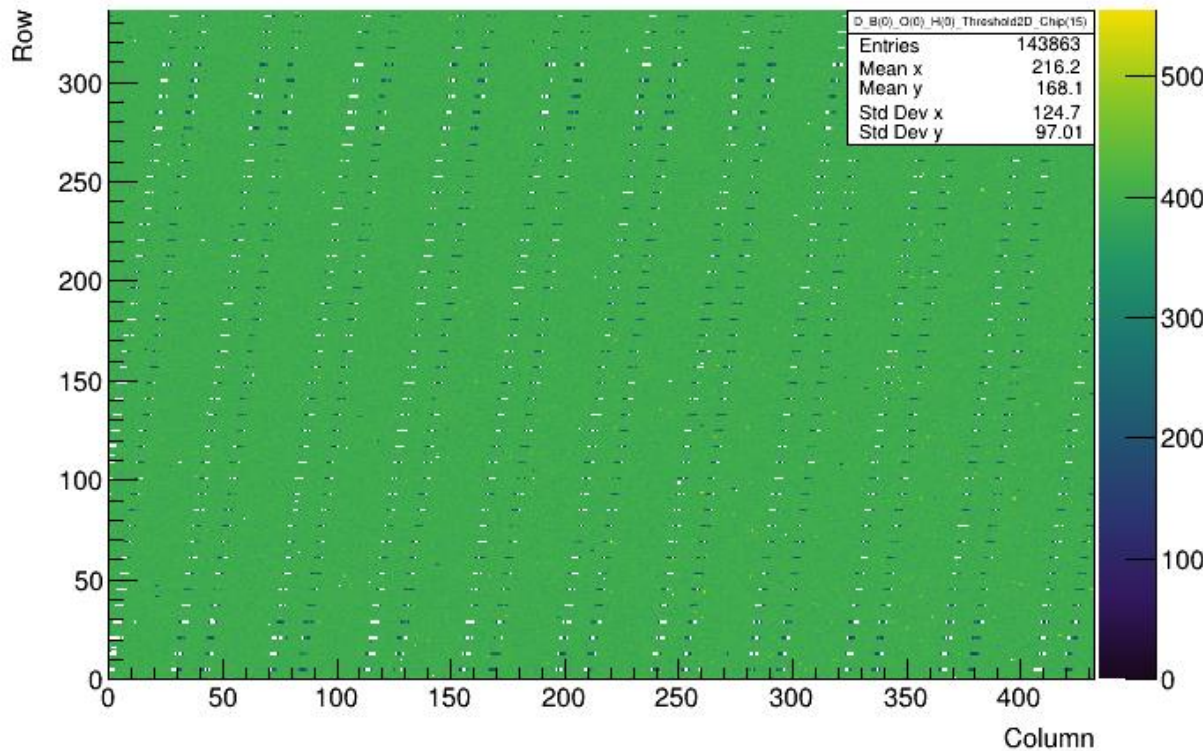


Injection pattern and faulty core columns

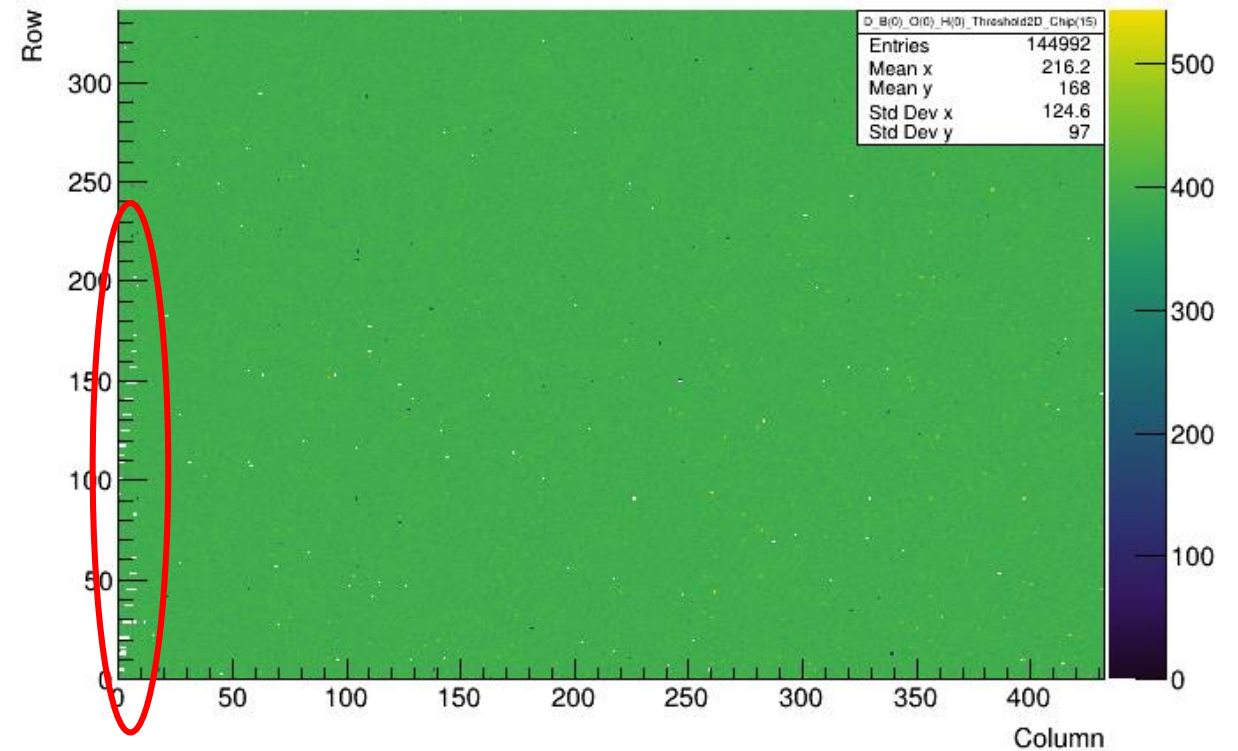
The faulty can be identified by looking at the intersection between the injection pattern and the faulty core column:

- By disabling them by hand, most pixels of the faulty core columns can be “saved”
- Just a fraction of the pixels belonging to columns [0,..,7] and the injection pattern must be disabled
→ most pixels can be **recovered** (useful e.g. to study big pixels)

D_B(0)_O(0)_H(0)_Threshold Map_Chip(15)



D_B(0)_O(0)_H(0)_Threshold Map_Chip(15)

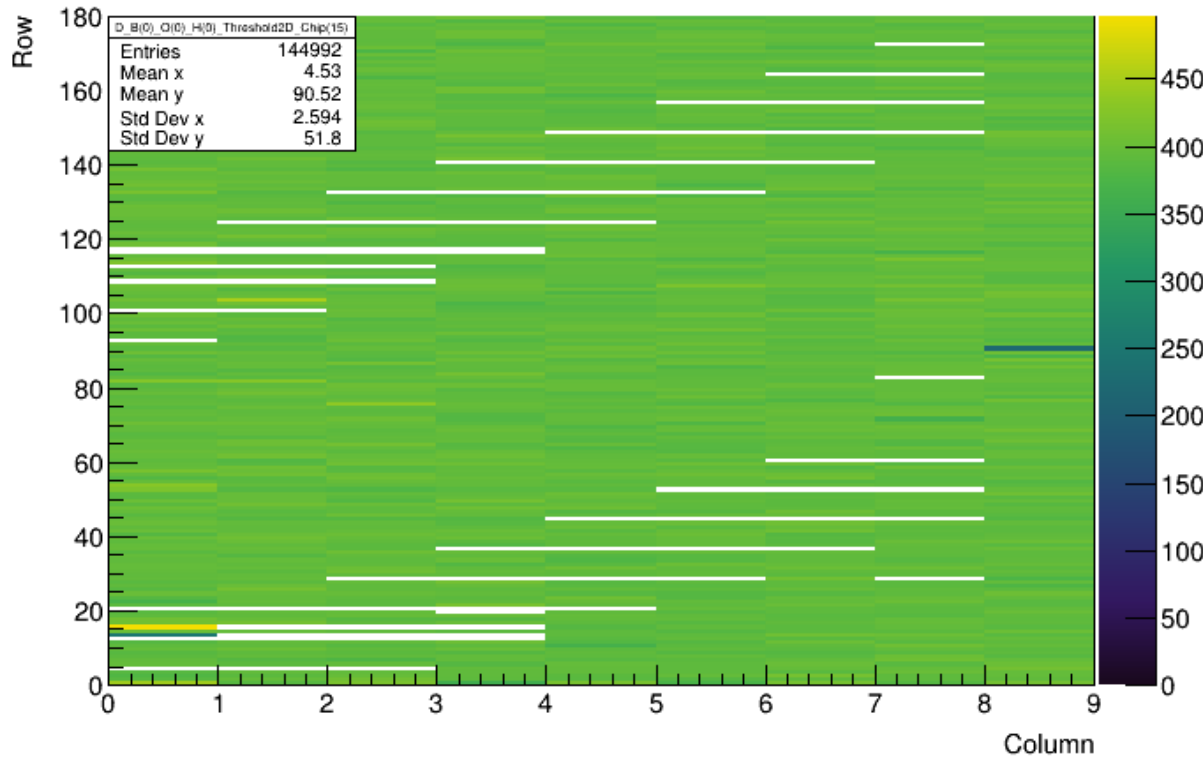


Injection pattern and faulty core columns

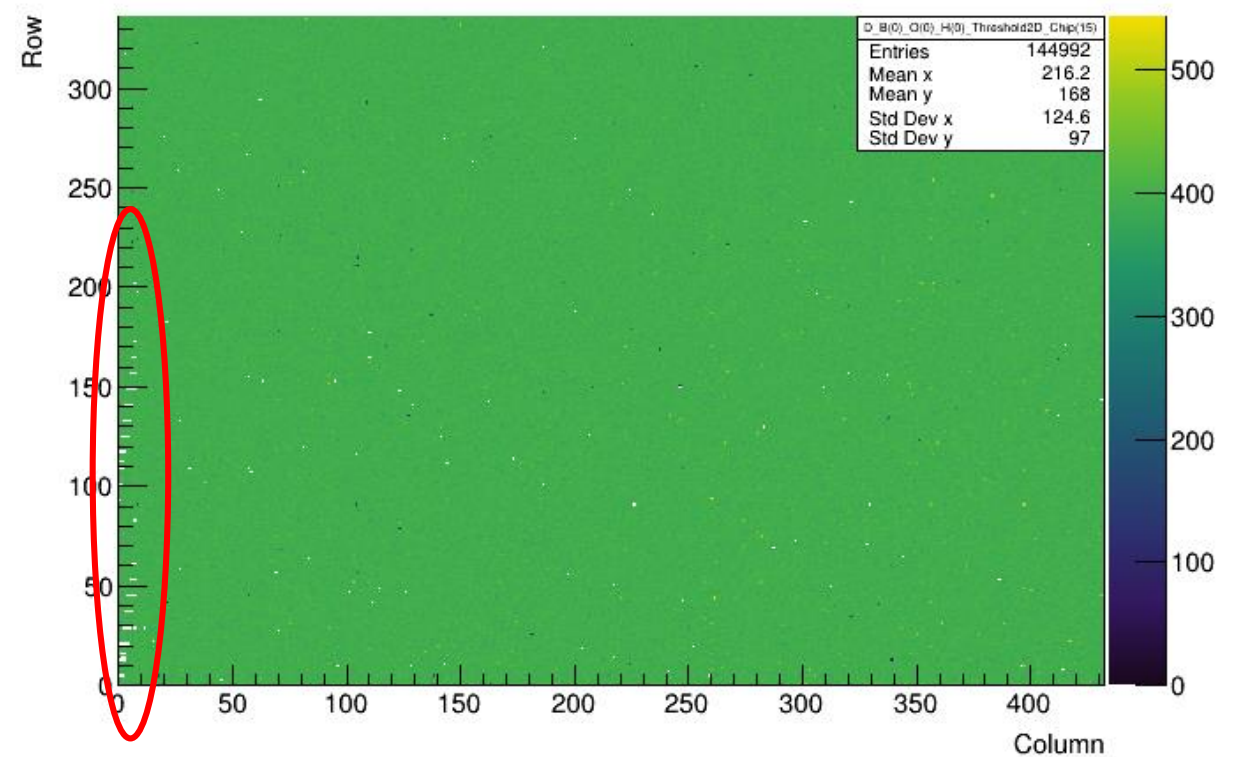
The faulty can be identified by looking at the intersection between the injection pattern and the faulty core column:

- By disabling them by hand, most pixels of the faulty core columns can be “saved”
- Just a fraction of the pixels belonging to columns [0,..,7] and the injection pattern must be disabled
→ most pixels can be **recovered** (useful e.g. to study big pixels)

D_B(0)_O(0)_H(0)_Threshold Map_Chip(15)



D_B(0)_O(0)_H(0)_Threshold Map_Chip(15)

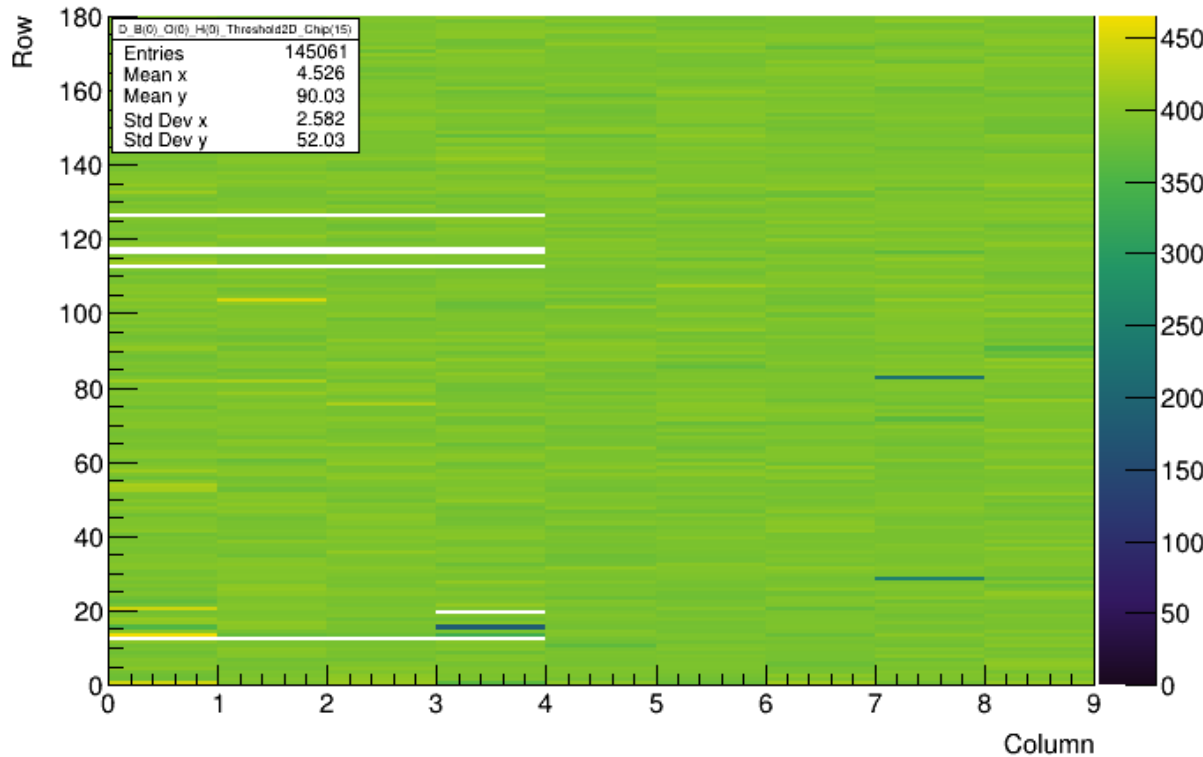


Conclusion

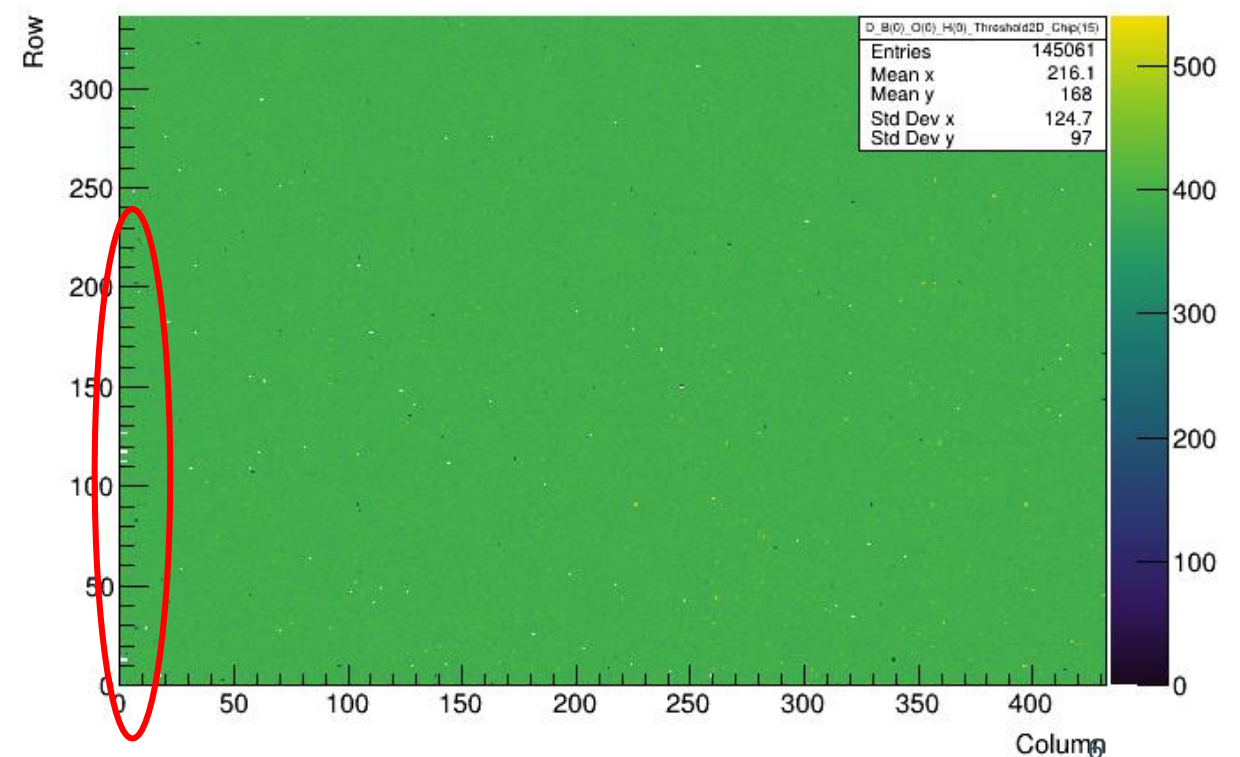
The faulty can be identified by looking at the intersection between the injection pattern and the faulty core column:

- By disabling them by hand, most pixels of the faulty core columns can be “saved”
- Just a fraction of the pixels belonging to columns [0,..,7] and the injection pattern must be disabled
→ most pixels can be **recovered** (useful e.g. to study big pixels)

D_B(0)_O(0)_H(0)_Threshold Map_Chip(15)



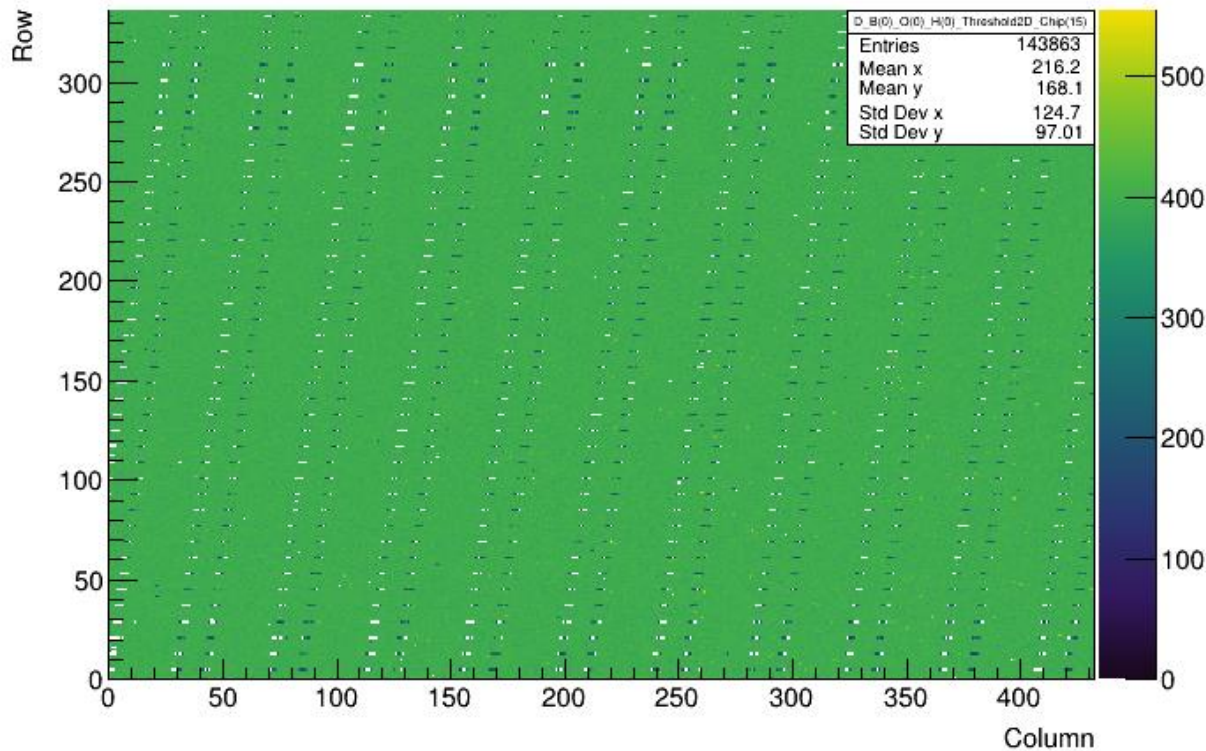
D_B(0)_O(0)_H(0)_Threshold Map_Chip(15)



Conclusion

- Outlook:
- Testing a **digital module** to check if the issue could be associated to the **flip-chipping procedure**
 - Implementing a pixel-by-pixel data integrity scan acting exclusively on the faulty core columns to automatically disable the faulty pixels

D_B(0)_O(0)_H(0)_Threshold Map_Chip(15)



D_B(0)_O(0)_H(0)_Threshold Map_Chip(15)

