

# Raw Banks

## SuperPixel Logic

Angelo Carbone, Serena Maccolini, Tommaso Fulghesu

Bologna

October 25, 2020

# SuperPixel

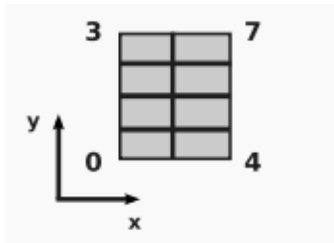


Figure: SuperPixel format

- 8 neighbouring pixels
- Orientation depends on the Sensor to which it belongs to
- First clustering techniques
- FPGA-friendly using Raw Bank

# Raw Bank

A raw bank contains information relative to each hit SuperPixel of a Sensor ( $192 \times 128$  SP)

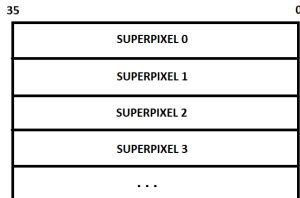
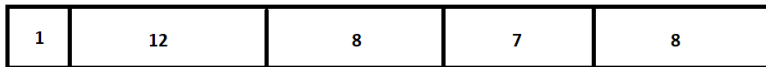


Figure: Raw bank format

## SP word

Each raw bank is composed by a 36-bit word



- 1 bits for the "hint"
- 12 bits for the SP time information (25ns sampling, 6ps time resolution)
- 8 and 7 bits for SP spatial position
- 8 bits for px inside SP

## SP Time distribution

Time coordinate associated to a superpixel is  $O(\text{ns}) \rightarrow 10$  bits

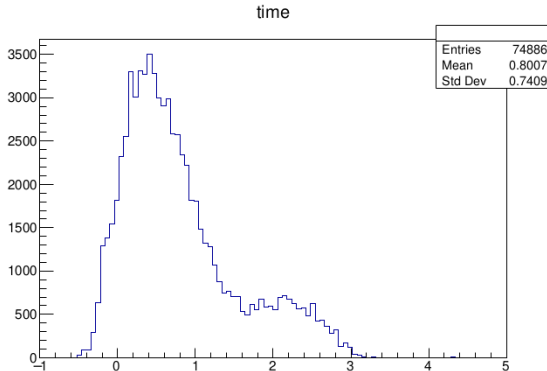


Figure: Time associated to a superpixel wrt event time

## SP time assumption

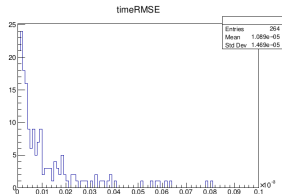
The time associated to each pixel is given by two terms:

$$t_{px} = t_{ov} + \frac{1}{v_{part}} * |z_{px} - z_{ov}| * \frac{p}{p_z}$$

**Assumption:**  $t_{SP} = t_{px_{first}}$

**TEST:** RMSE for each SP

**RESULT:** Few number of SPs with more than 1 hit pixels have RMSE  $\neq 0$   
(264 entries vs 74886 SP)



**Figure:** RMSE  $\neq 0$  associated to a superpixel with more than one ON pixel

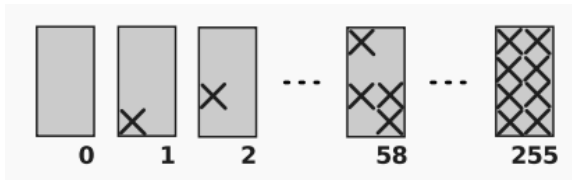
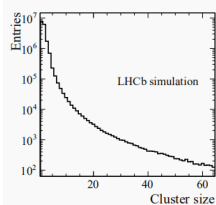
## Next steps

- Test directly with FPGA;
- Build raw bank even using pixel logic;
- Clustering algorithms based on the SP.

# FPGA Clustering

WHY?

Large amount of clusters are inside an isolated SP  $\rightarrow$  It is possible to use a look-up table (from 0 to 255) to see the active pixels inside the SP and creates the clusters (2x faster)

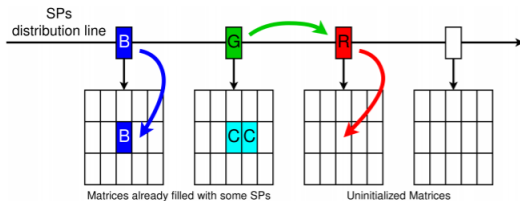




# FPGA Clustering

## HOW IT WORKS?

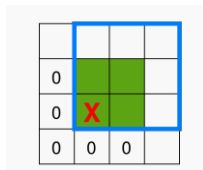
STEP 1: Matrices with dimension  $5 \times 3$  SPs ( $10 \times 12$  pixels) at every clock cycle change SP input.



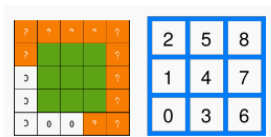
# FPGA Clustering

## HOW IT WORKS?

STEP 2: Construction of cluster candidate from the **SEED** pixel.



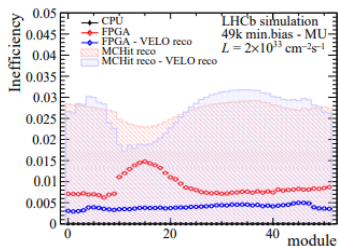
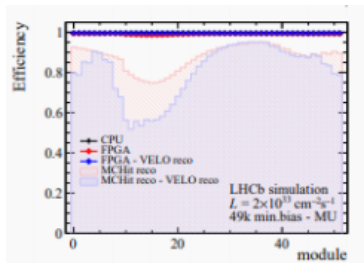
## TOPOLOGY



To every seed pixel is associated a lookup table with *flags* which characterize the cluster

Meaning	Flag
Isolated	101
Overflow	100
Self-contained & edge	011
Self-contained & not-edge	010
Not-self-contained & edge	001
Not-self-contained & not-edge	000

# Clustering efficiency

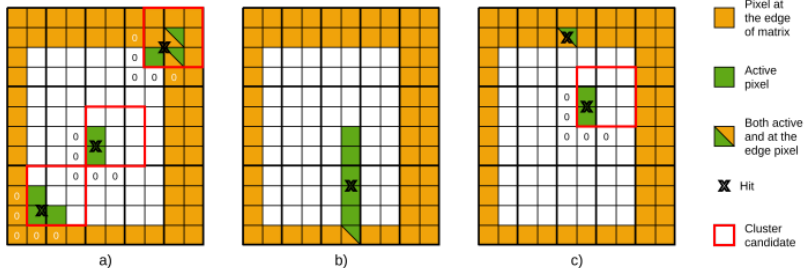


$$\epsilon = \frac{N_{MC_{linked}}}{N_{MC}}$$

$N_{MC_{linked}}$  = # hits with linked reconstructed cluster.

$N_{MC}$  = # reconstructible hits

# Cluster Inefficiency



FPGA efficiency depends on VELO occupancy

- larger prob. of non isolated SPs and larger cluster dimensions
- larger prob. of overflow