Progress on 4D Tracking R&D in ATLAS experiment

Hongtao Yang (USTC) on behalf of all the contributors

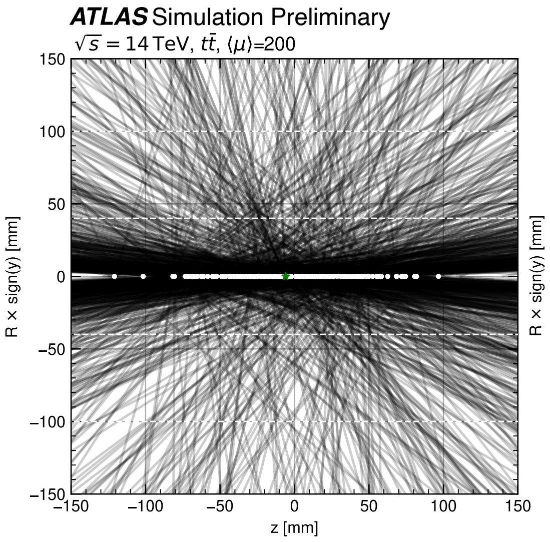
2026年超级陶粲装置研讨会
July 3, 2026

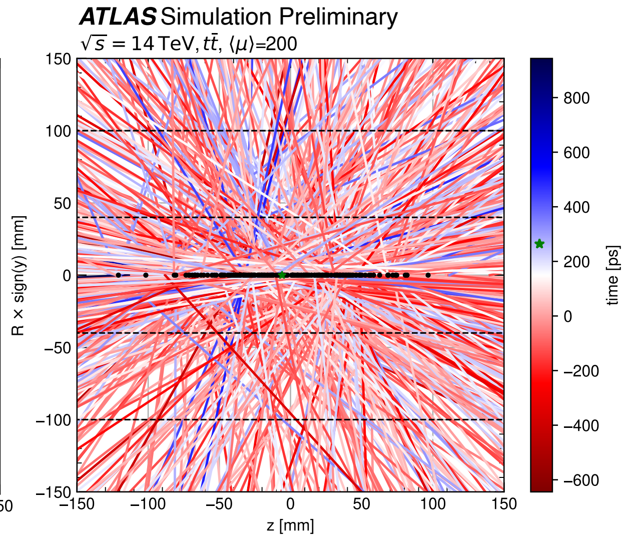
Pictures from: ATL-PHYS-PUB-2023-023: Investigating the impact of 4D Tracking in ATLAS Beyond Run 4

Introduction
Chance of future 4D tracking for ATLAS ITk

- Motivation – LHC upgraded to HL-LHC

  Luminosity up to $7.5\times10^{34}\ \mathrm{cm}^{-2}s^{-1}$, pile-ups up to **200**

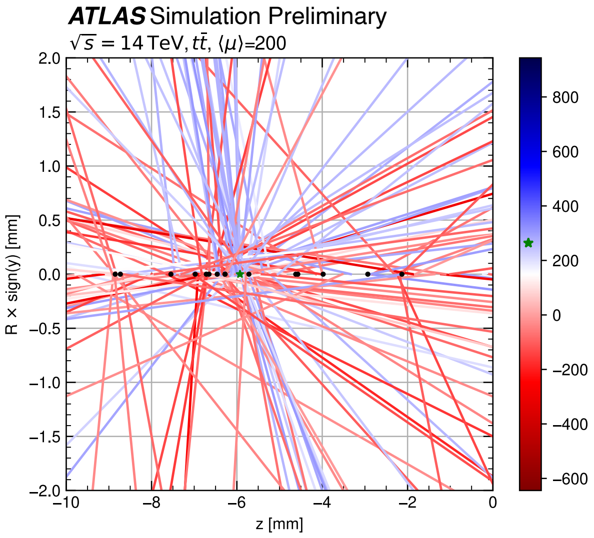




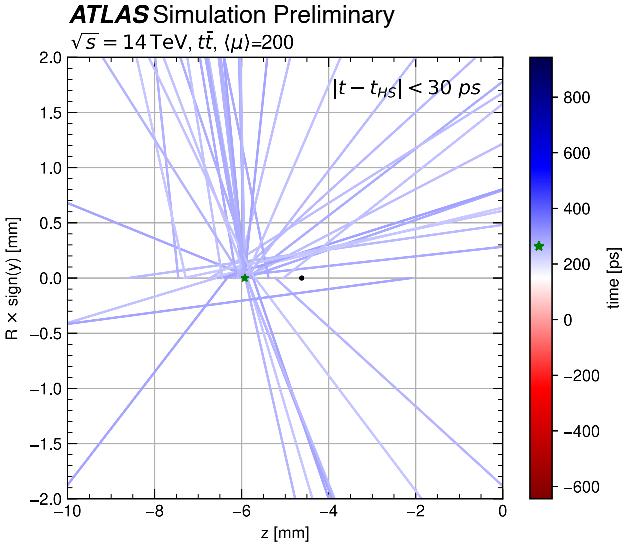
Longitudinal view of a simulated $t\bar{t}$ event with <μ>= 200 in ATLAS ITk

No time information

Add time information



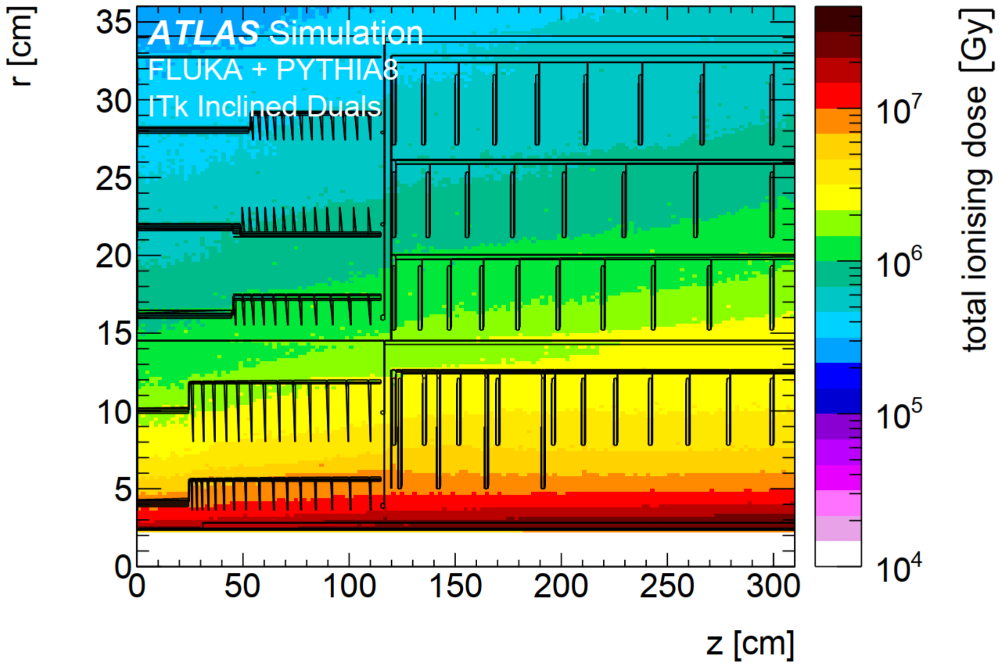
Zoom in space



Separation by time

Introduction
Chance of future 4D tracking for ATLAS ITk

- ATLAS ITk Pixel only has traditional 3D tracking and vertexing capability
  - The two innermost layers of the ATLAS ITk Pixel detector are designed for a maximum luminosity of **2000 fb⁻¹**
  - ATLAS High Granularity Timing Detector (HGTD) also provides timing, but only 1 year in endcap out of ITk volume
- Replacement of ITk Pixel inner layers ATLAS in the middle of HL-LHC operation provides a chance to implement **4D tracking & vertexing**
- This talk reports the latest progress of **4D tracking algorithm and detector R&D**



The total ionizing dose distributions for the ITk Pixel Detector.
<5 MGy in 4000 fb^{-1} for unreplaceable part
up to 10 MGy for 2000 fb^{-1} for Layer 0

[ATLAS-TDR-030 Inner Tracker Pixel Detector](#)

4D tracking with ACTS

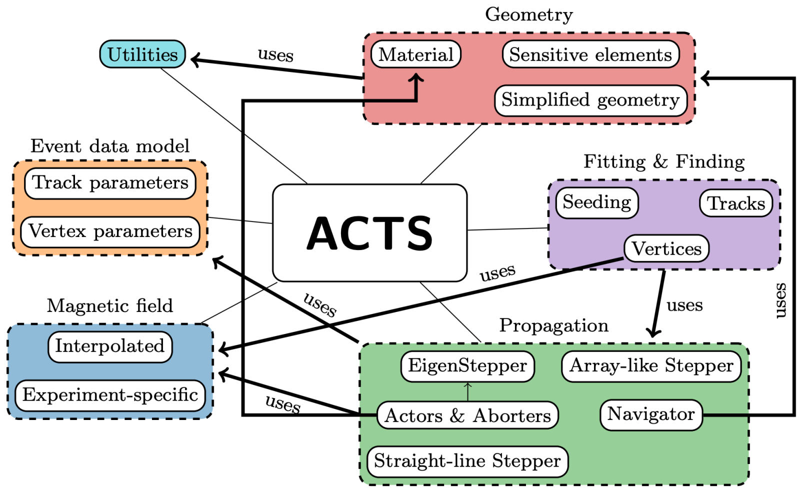
ACTS GitHub: https://github.com/acts-project/acts
ACTS docs : https://acts.readthedocs.io/en/latest/index.html

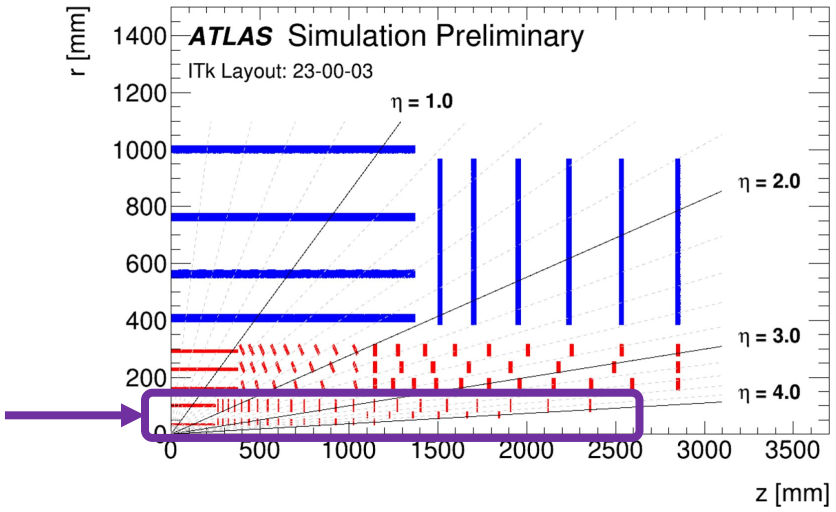
Introduction to Software

- **ACTS** (A Common Tracking Software) is an general toolkit for charged particle track reconstruction in particle physics experiments
  - A track in ACTS is described by $\vec{x} = \left(l_0, l_1, \phi, \theta, \frac{q}{p}, t\right)$: time is naturally included
- Smear truth hit information in the ITk innermost 2-layer pixel region with gauss distribution

$$\sigma(t) = 50\,ps$$

as a rough estimation of hardware performance





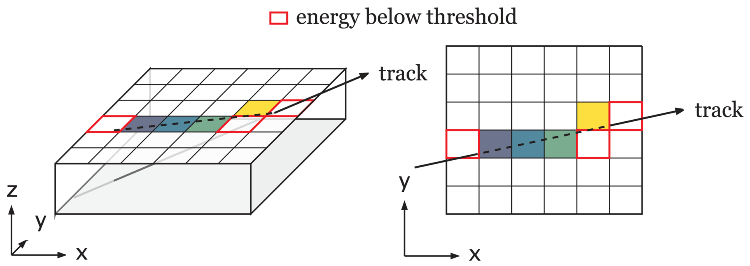
Full chain workflow in ACTS

with time algorithm

propagate time information

Fatras / Geant4

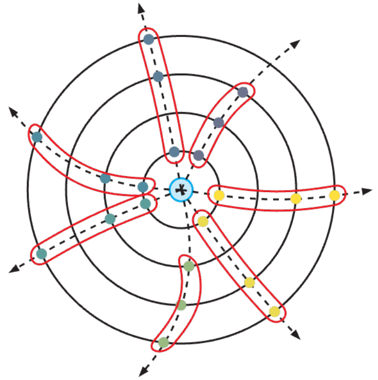
simulation



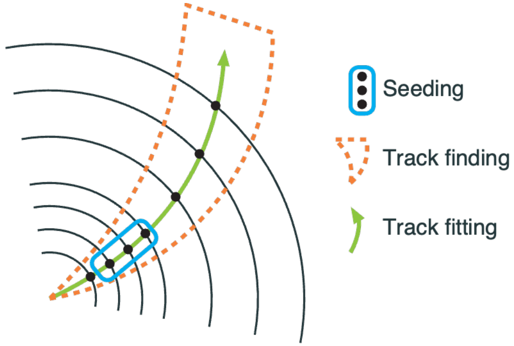
digitization & clustering

time information

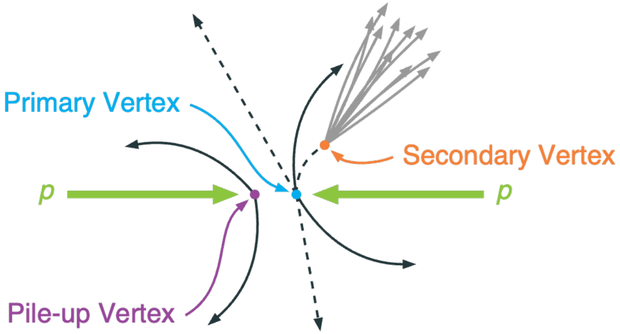
use smeared hit information



seeding

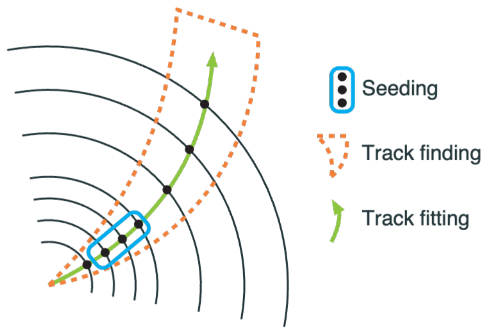


track finding& fitting (CKF)

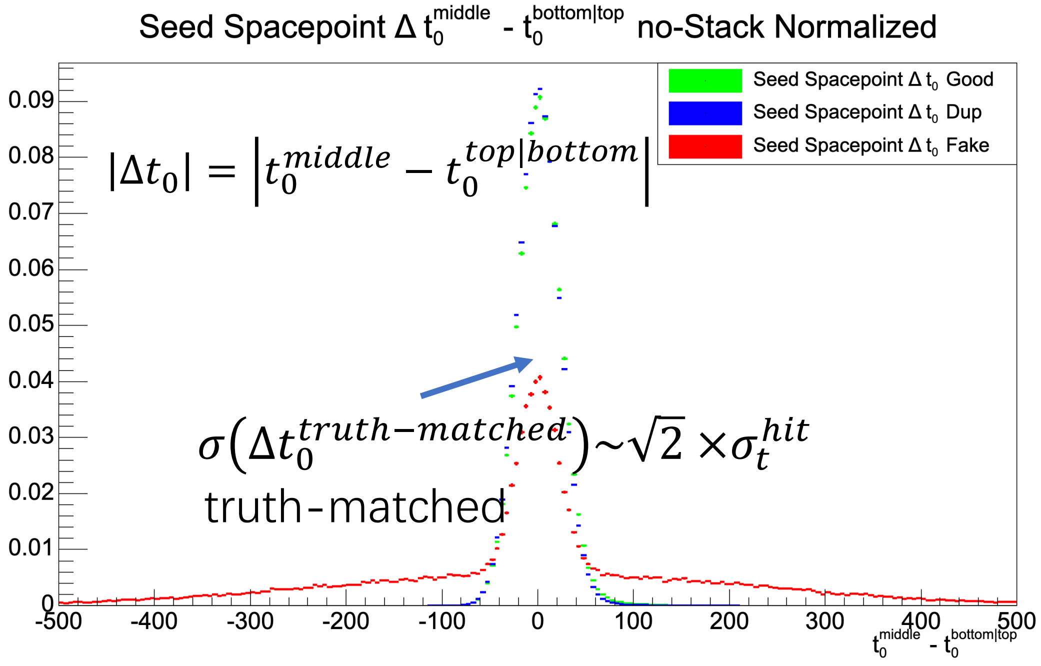


vertexing

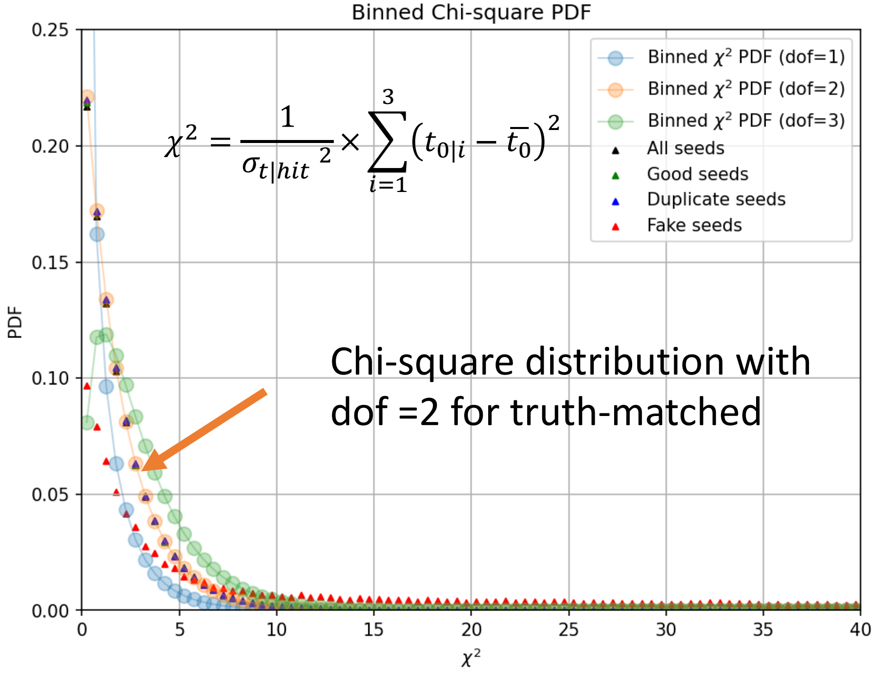
4D seeding algorithm



- Variable t_0 for spacepoint $$t_0 = t_{arrival} - \frac{1}{c} \cdot \sqrt{x^2 + y^2 + z^2} = t_{arrival} - t_{fly}$$
- A gauss filter for **doublets** & a chi-square filter for **triplets**

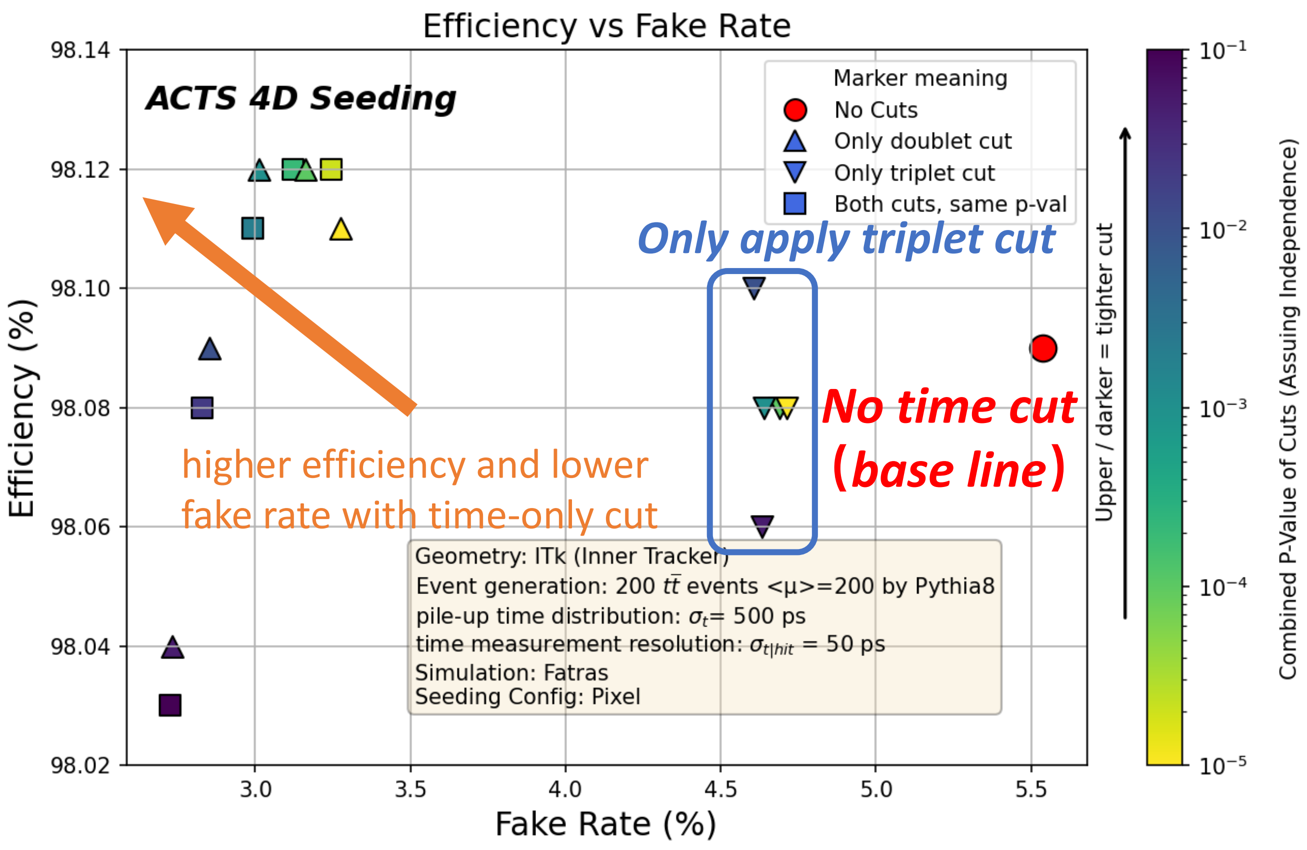


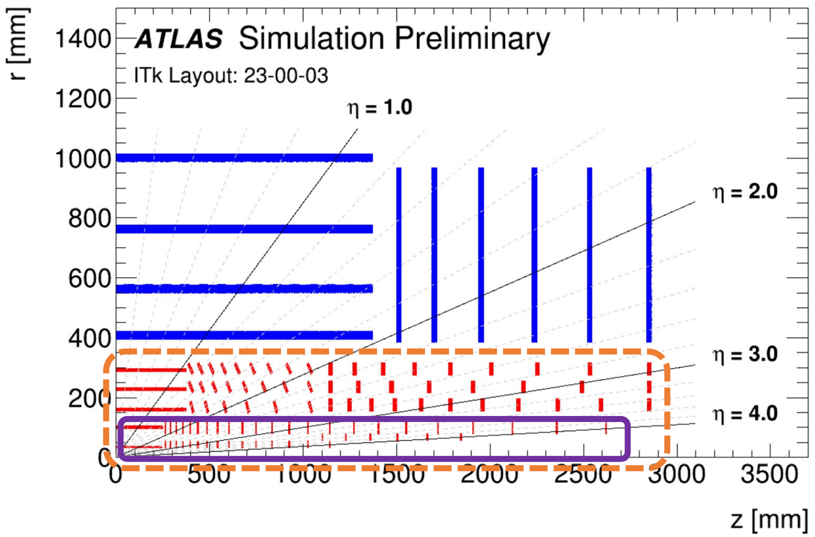
Doublet - Normalized distribution of Δt_0



Triplet - Normalized distribution of χ^2

4D seeding performance



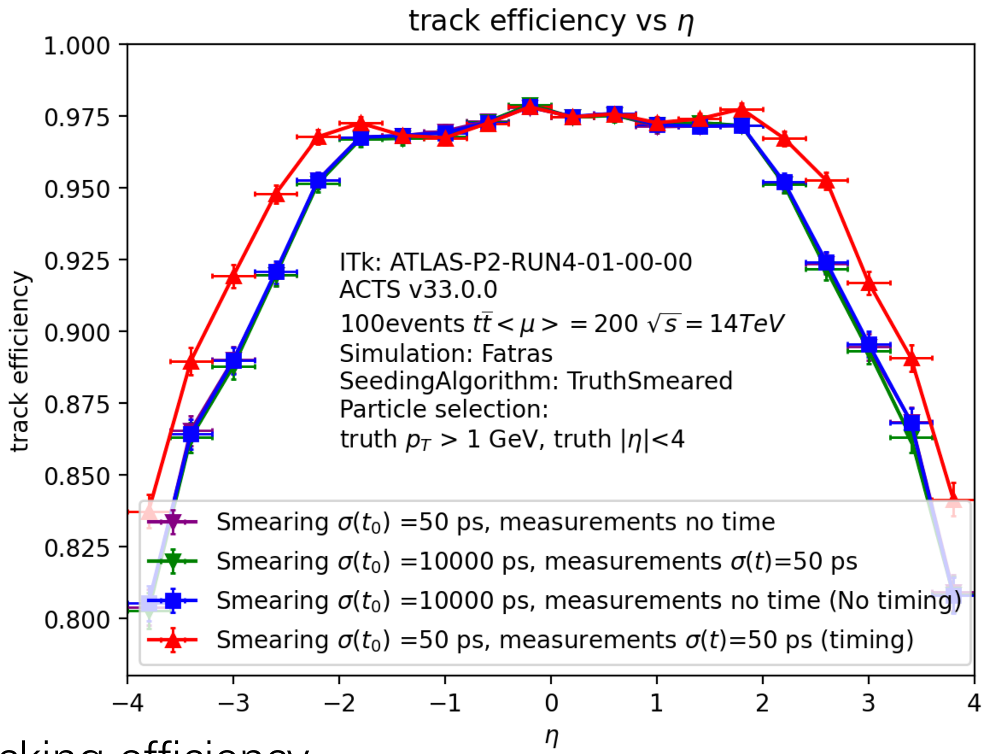


- Using **pixel (5 layers)** spacepoint seeding
- Adding time for innermost **2** layers

p-value: corresponding to the probability of excluding a truth-matched seed by cut

Doublet-cut seems more effective on this condition

4D tracking performance



Tracking efficiency

93.72% (no timing) to 94.87%↑ (50 ps timing)

93.71% (only “measured” time for initial track parameters)

93.63%↓ (only time for measurements)

Combinatorial Kalman Filter $\vec{x} = \left(l_0, l_1, \phi, \theta, \frac{q}{p}, t\right)$

$$\vec{x}_k = \vec{x}_k^{k-1} + \mathrm{K}_k(\vec{m}_k - \mathrm{H}_k \vec{x}_k^{k-1})$$

- Add time for both **initial track parameters** and **measurements** of innermost 2-layer pixel
- Event generation: ***$t\bar{t}$* events** $\langle\mu\rangle$=200 with Pythia 8
- **Fatras** simulation & **Truth-Smeared** seeding

RED (4D tracking): CKF initial track parameters and measurements have time information

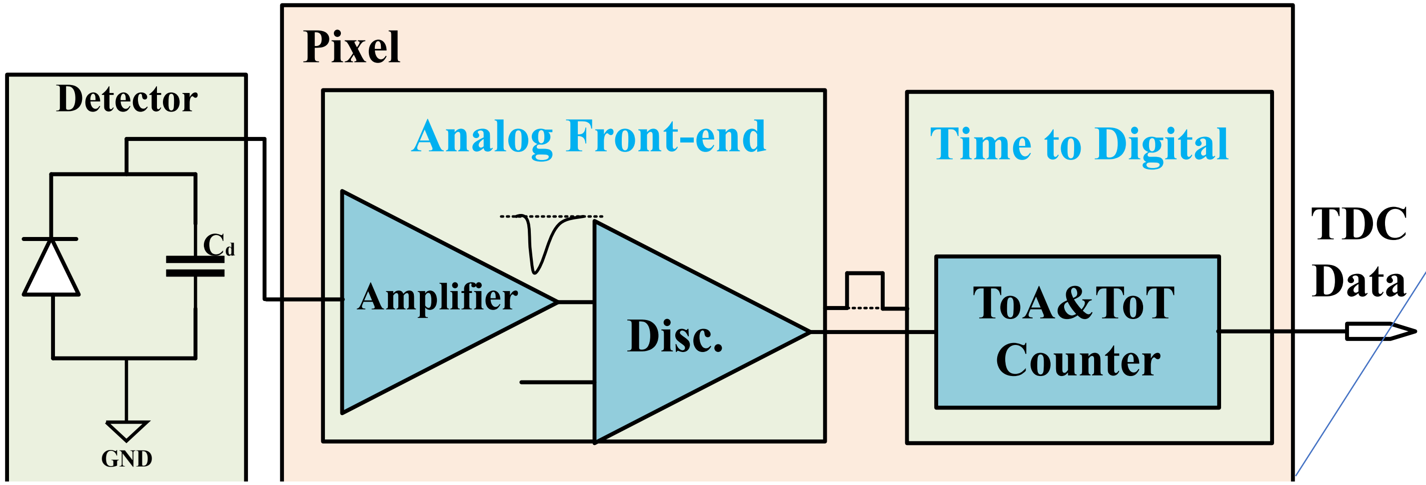
BLUE (3D tracking): no time information for CKF initial track parameters and measurements

Timing Pixel ASIC design

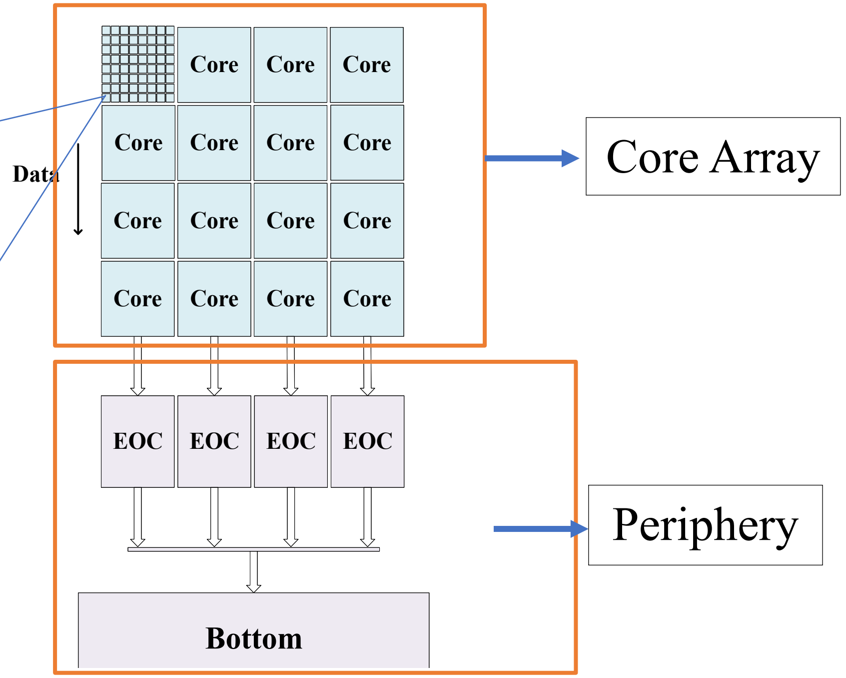
Overview

➢ **Modules** for the Readout ASIC:

- **Pixel**: Amplifier, Discriminator, Time-to-digital converter, 200 μm×200 μm
- **Cores Array**: 1 core includes 8×8 pixels, same as ITKPix. 8 pixels are in one **region**.
- **Periphery**: End of Column, Chip Bottom.



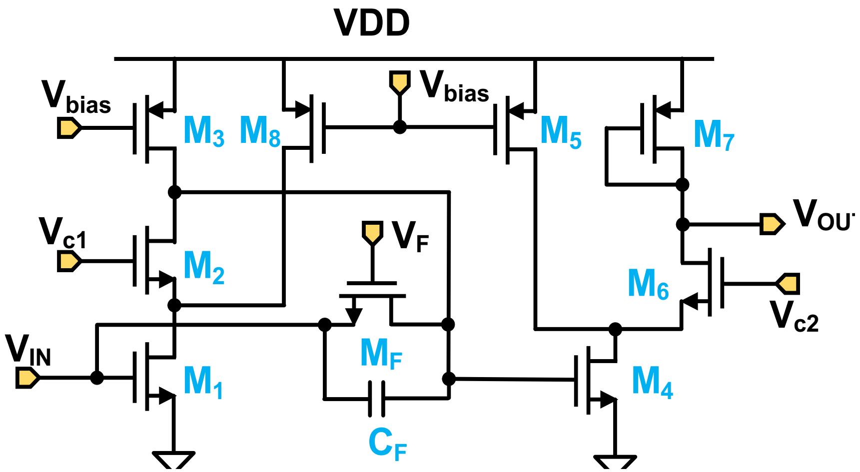
Pixel circuit



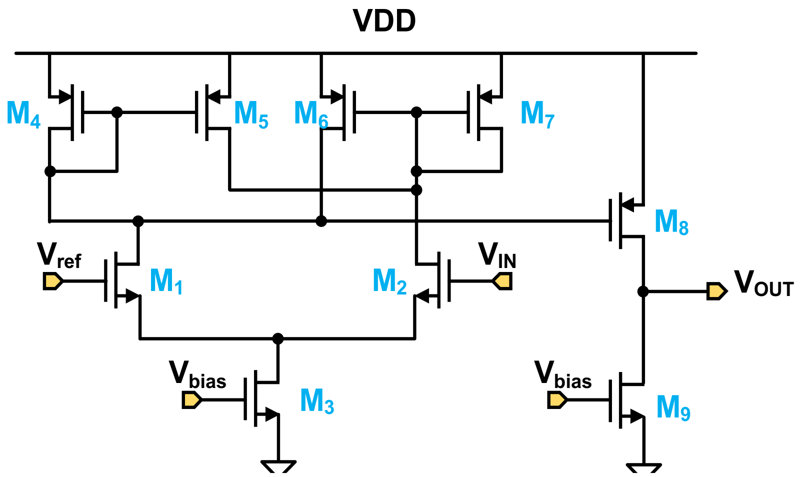
Overview

Pixel: Analog Front-end

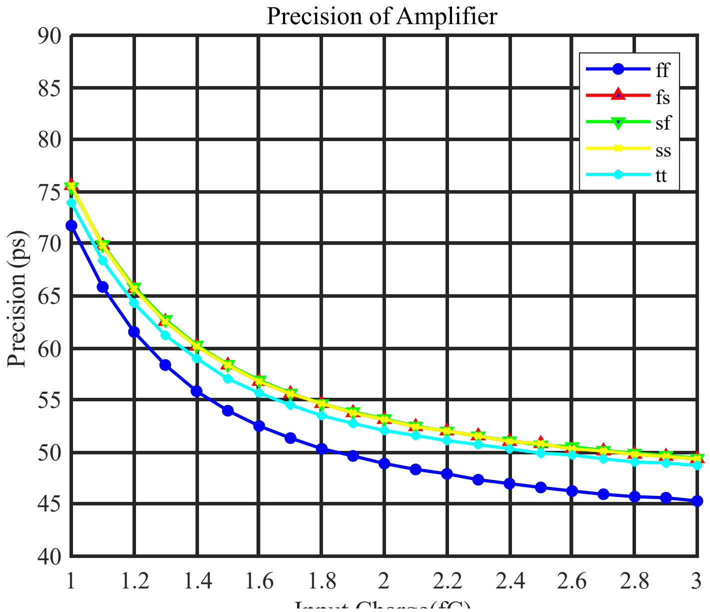
- **Two-stage Amplifier**:
  - Charge sensitive amplifier: folded cascade + Voltage Amp.
- **Two-stage Discriminator**:
  - Differential driver: positive feedback
  - Common source driver: saving power
- **3D sensor@3fC**



Amplifier



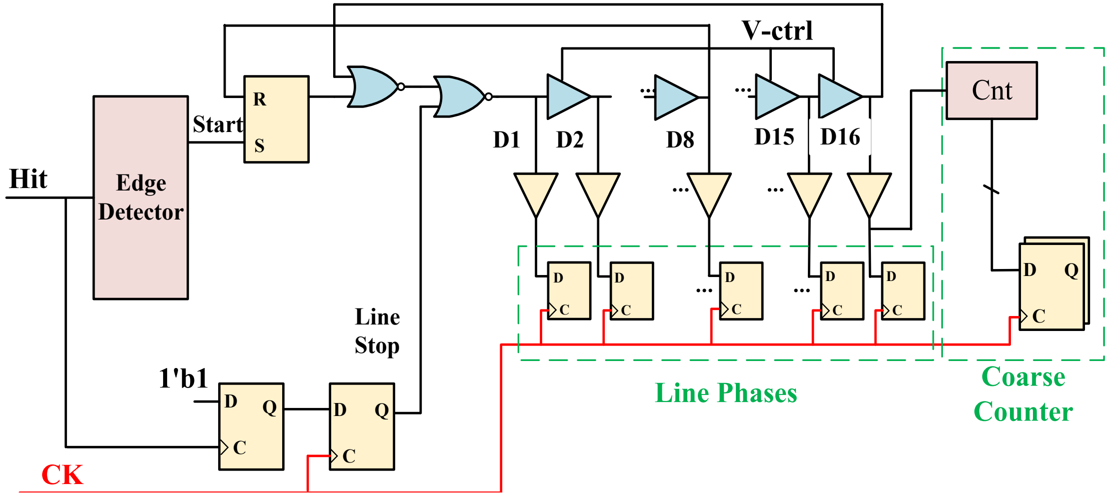
Discriminator



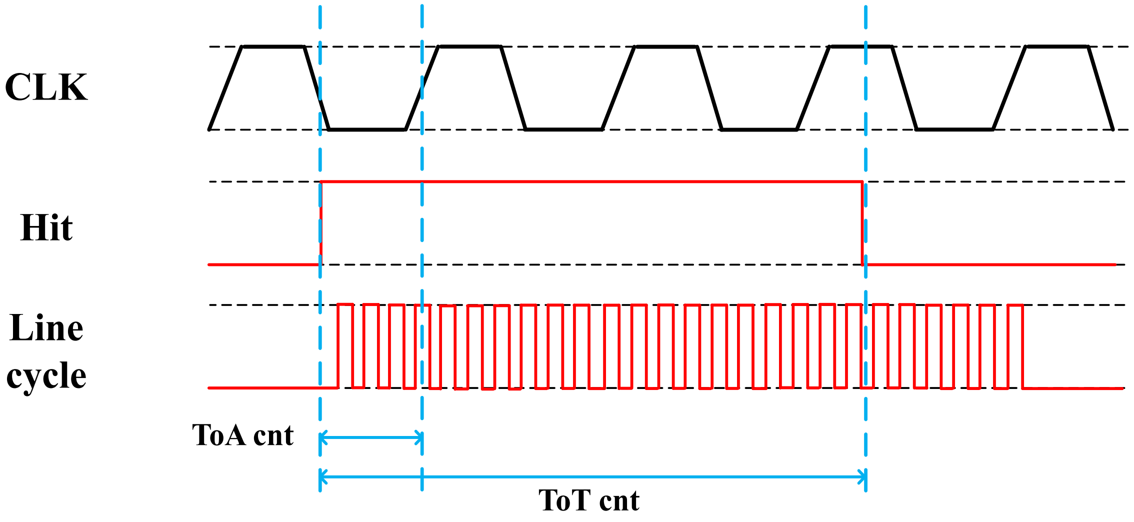
Precisions of Amp. + Discr. (130nm)

Pixel: Time to Digital Converter

- **Delay line + Loop counter:**
  - Edge detector
  - SR latch
  - ToA & ToT measurement (bin size ~ (90~120) ps)



Designs for Time to Digital Converter

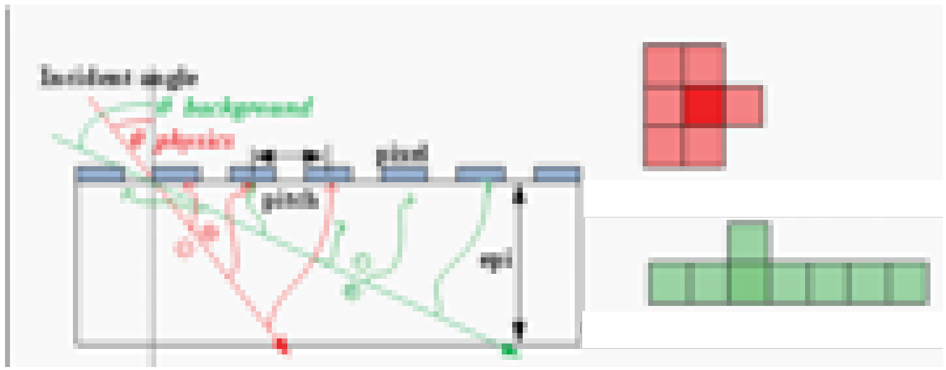


Designs for Time to Digital Converter

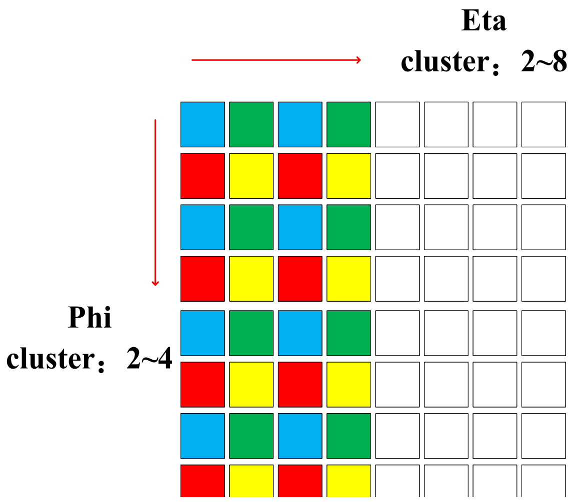
- ToA measurement:
  - Line's cycle + Line's phase
- ToT measurement
  - Only line's cycle

Core

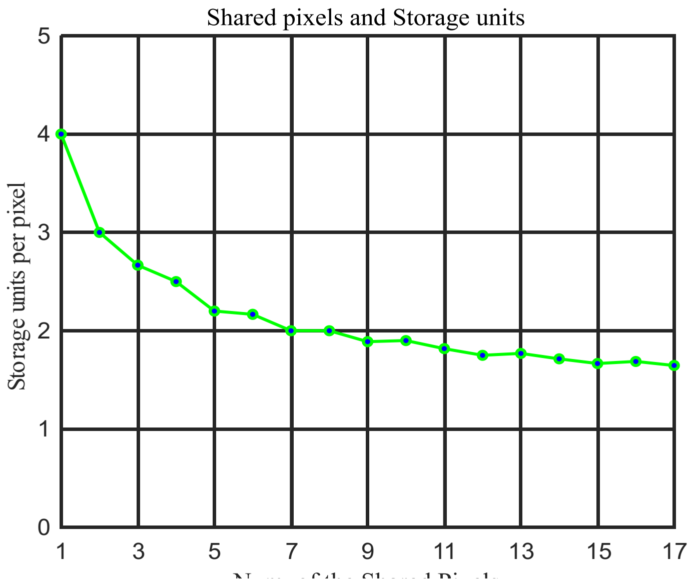
- Shared memory structure:
  - Reduce the storage space due to the sparse hits
- Considerations:
  - All pixels shared need to be independent.
  - The cluster size effect is universal in the detector.
- Eliminating the cluster effect:
  - Using a specific map to share the pixels



The cluster size



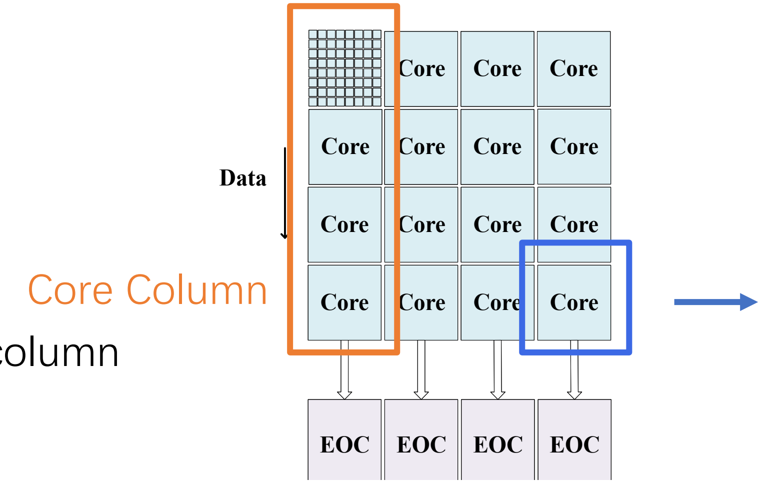
Shared map



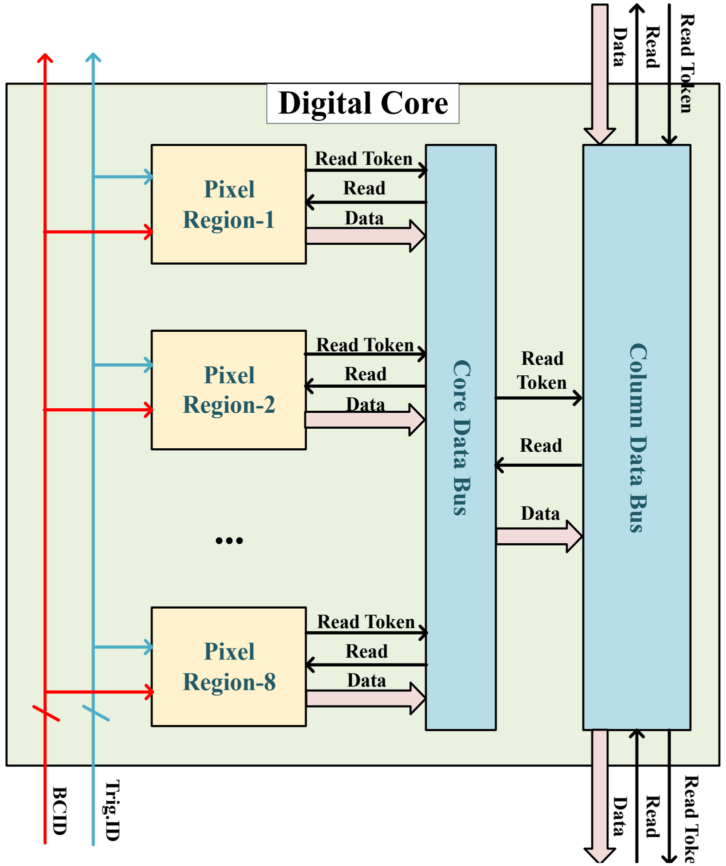
Storage space

Core

- Core degin:
  - 8 × 8 pixels, core is same as others and they are connected as a chain
  - Receiving and buffering data from TDC
  - complete the trigger matching
- Shared memory structure:
  - 8 pixels shared in one **region**
  - One core consists of 8 regions
- Readout:
  - Priority readout
  - Core column readout to the End of column



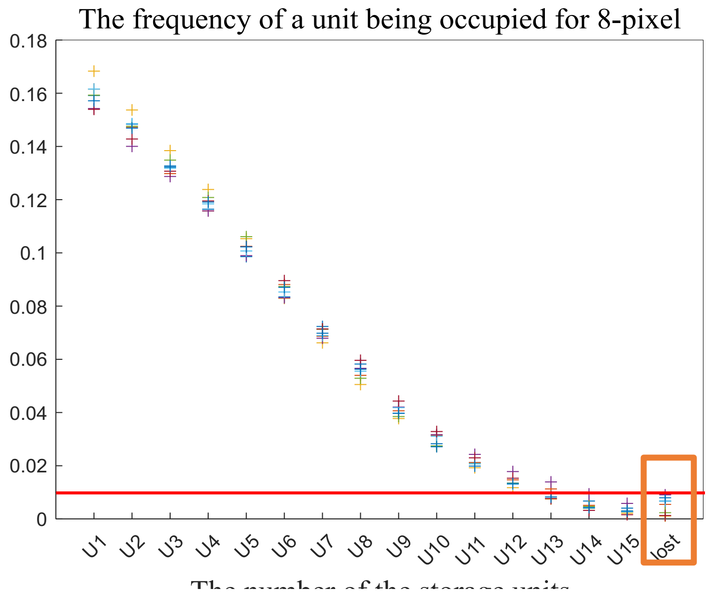
Core array



Core design

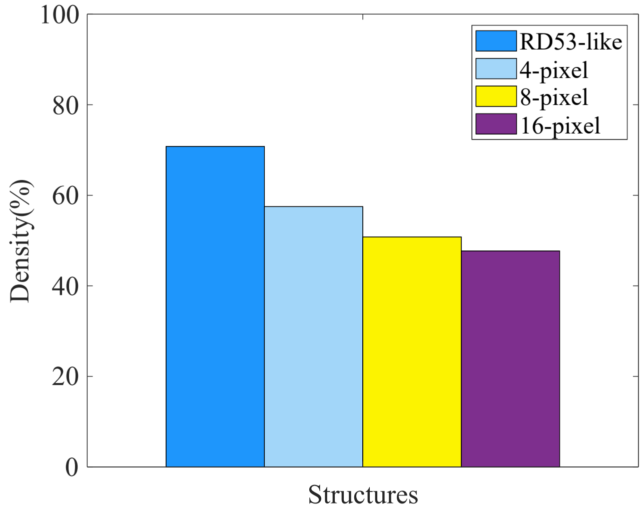
Simulation

- The simulation results (in 130 nm)
  - Holding the corresponding data rate (~75 kHz/pixel)
  - Lower power, Lower density.
  - Shared memory could reduce power and area obviously.

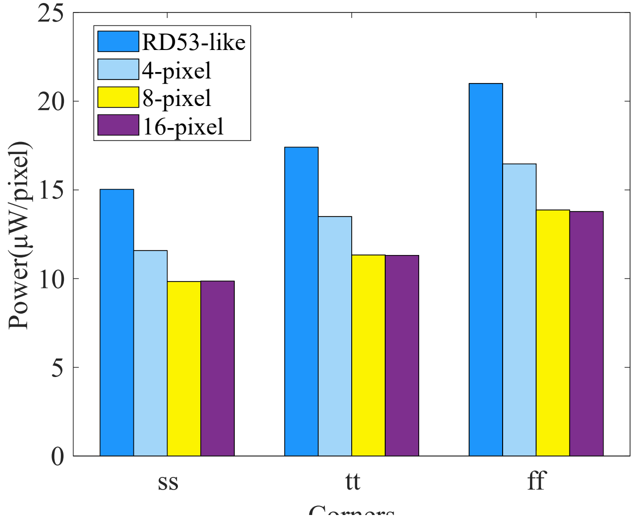


Lost rate lower than the red line (1%)

The performance of shared memory (eg.8-pixel)



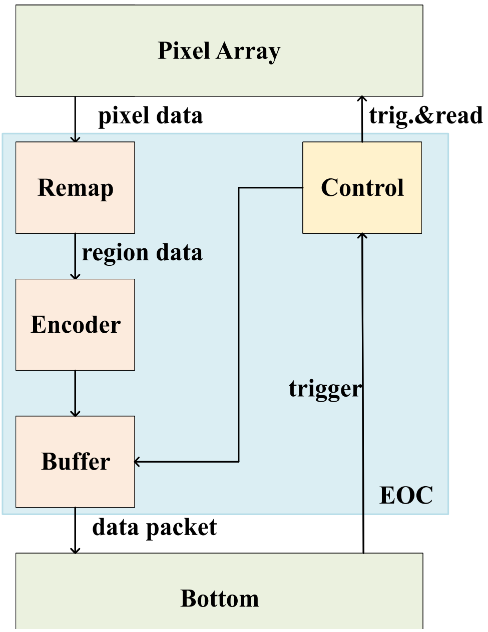
The area comparison for the layout



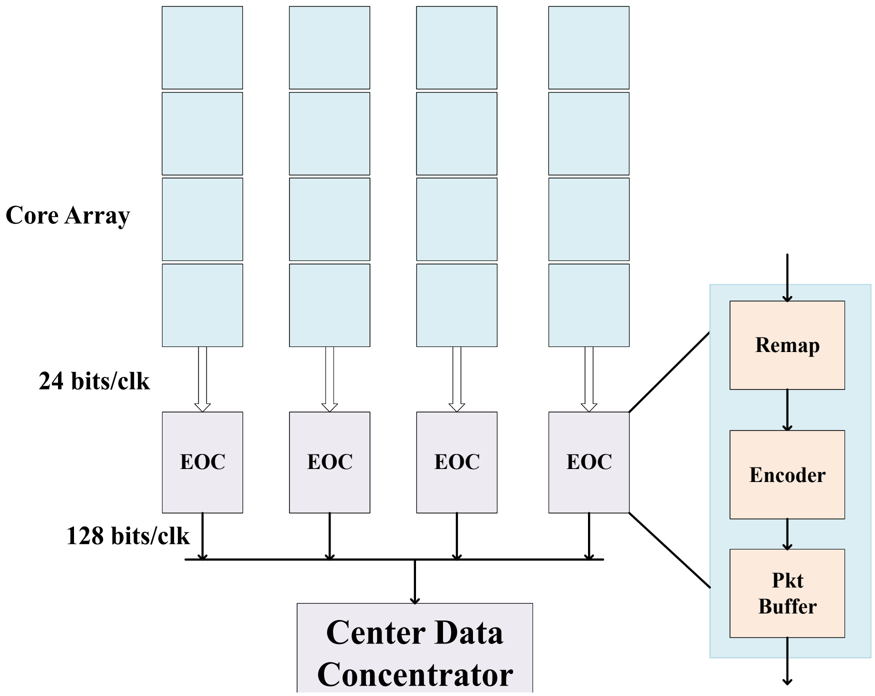
The power comparison for the layout

Periphery

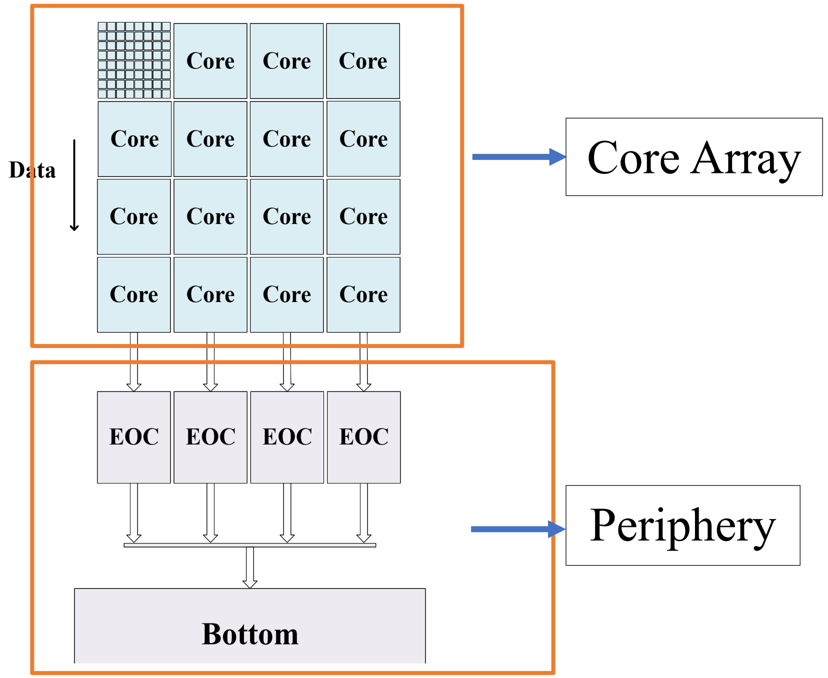
- Modules for the Periphery:
  - **End of Column**: packaging and encoding and compressing the data.
  - Chip Bottom: concentrating the chip data and control the trigger.



EOC design



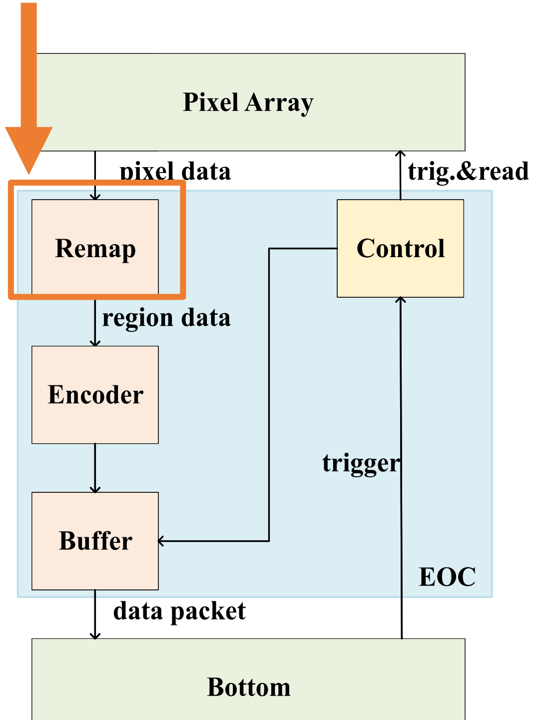
Data flow



Overview

End of Column

Remap data to make it more concentrated



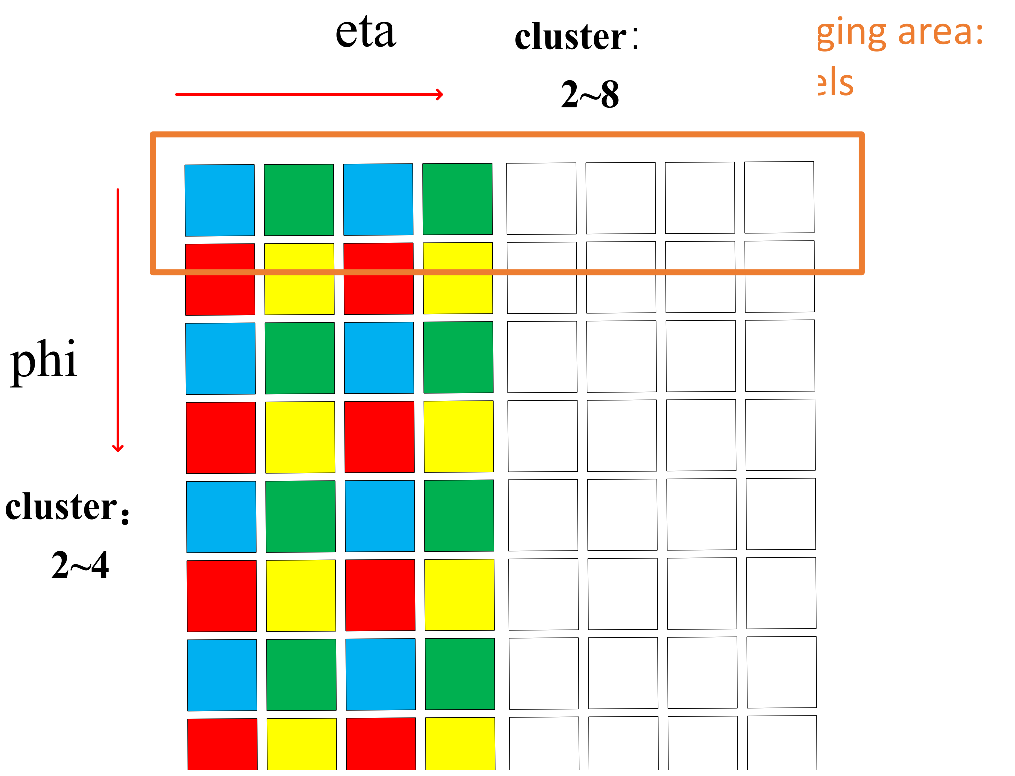
EOC design

End of Column:

- ➢ controlling the readout of core column, and packaging the hit data

Shared memory:

- ➢ **Disrupt** the order of reading data from the core.
- ➢ Breaking up the concentration of clusters. Making data more dispersed which is **not conducive** to data compression

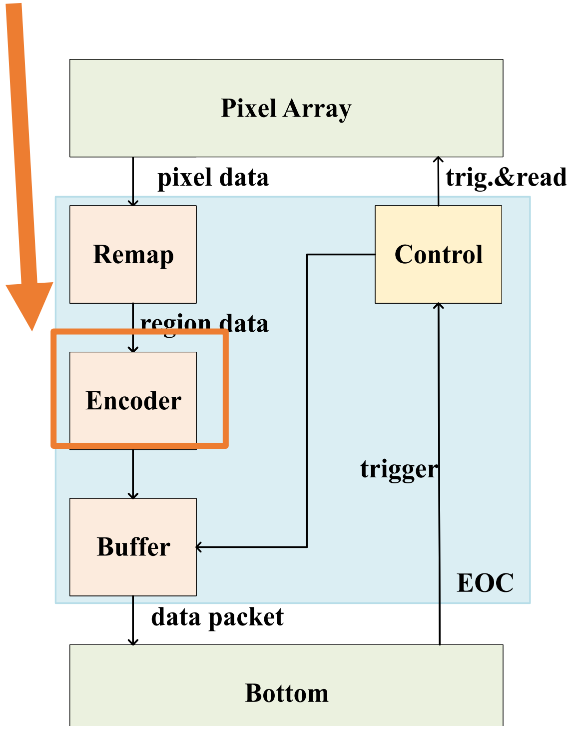


Remap module:

- ➢ **Reorganizing** the hit as the packaging area.
- ➢ The packaging area use the cluster effect to concentrate the data.

End of Column

Hoffman coding to compress



EOC design

Packaging data: coarse address + hit map + hit data

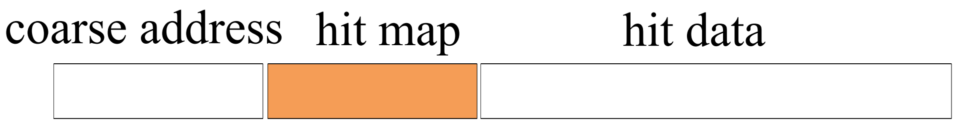
Encoder:

- Based on the frequency distribution of hit-map in simulation data, use Huffman encoding to compress the hit-map

Encoding schemes:

- Comparing several compression schemes to get the best
- Compression efficiency:

  packaging + Huffman → ~73% (27% reduced)

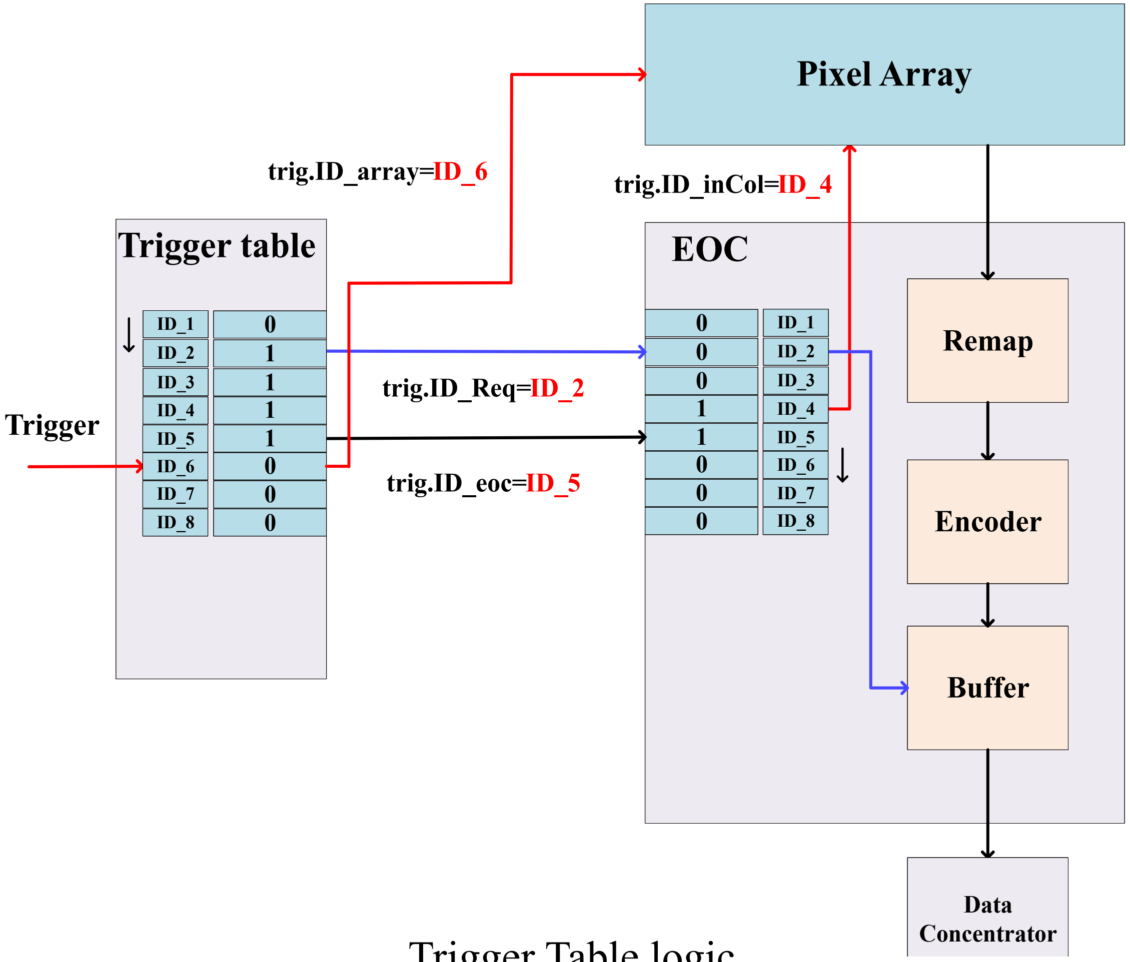


A data packet

Encoding of hit map		Entropy
Bitmask		8
Binary-tree		7.79
Independent address		11.20
Multi-stages Hoffman	4-6-14	7.17
	3-6-14	7.84
	6-14	6.34

Encoding schemes

Bottom



Trigger Table logic

- Functions for the Bottom:
  - **Trigger table**: manage the trigger and generate the trigger ID to control the readout.
  - **Data concentration**: concentrating the chip data.

Summary

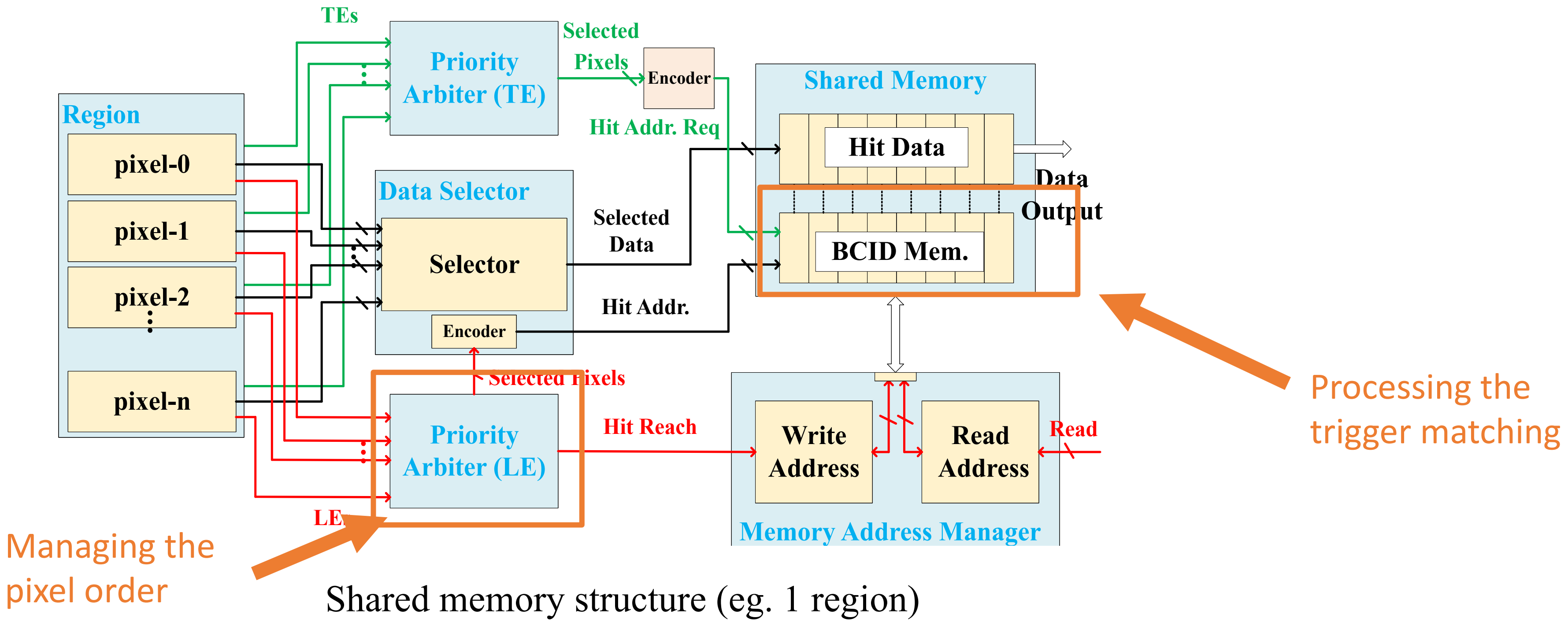
- Full 4D tracking workflow implemented in ACTS as well as ATLAS Software Framework
  - Next step is to carry out physics case study: b-tagging, HH search
- Timing Pixel ASIC design in good progress
  - First 130 nm prototype ASIC expected to be fabricated this year
  - 3D sensor R&D has started

Thank you for your attention!

Backup

Region

- Functions:
  - Storing the time stamp (BCID) and TDC data, doing the trigger matching.
- Two **priority arbiters** for Leading Edges and Trailing Edges:
  - Manage the address of memory.
- Every cell in shared memory needs a FSM to process **trigger matching**.

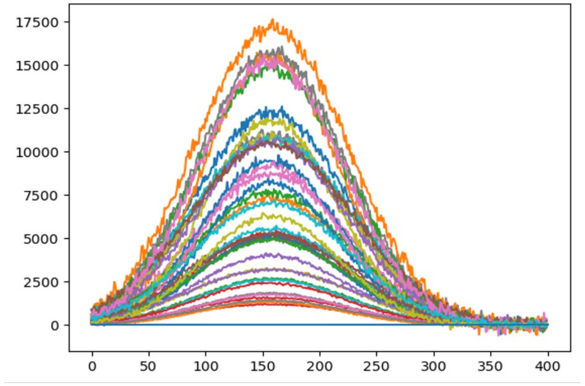


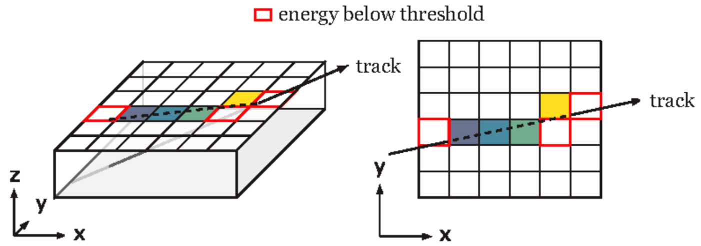
Shared memory structure (eg. 1 region)

Timing pixel digitization in Athena

➢Digitization

- simulate detector response from hit simulation inform
- HIT (Geant4 simulation) -> RDO (Raw data output)
- (position, charge...) -> (identifier/position of cells, ToT, ToA)





Graphs from ACTS docs

Geant4 hit time

smeared time
(add random gaussian distribution)

simulate pulse and calculate ToA
based on parameterized pulse waveform

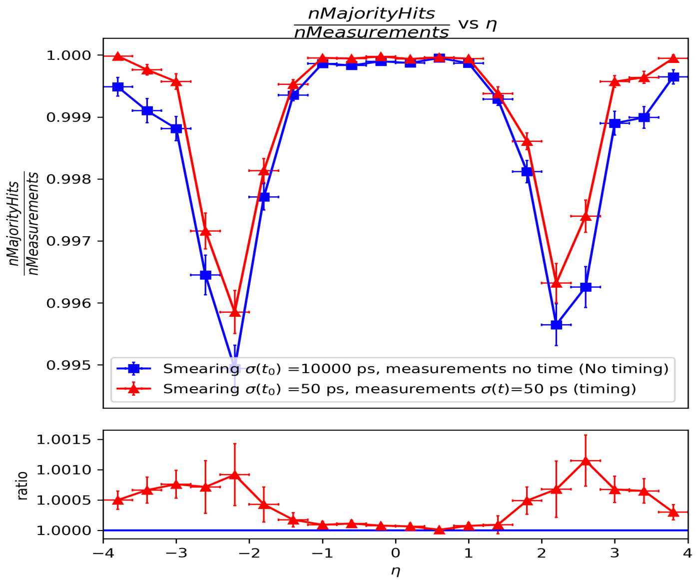
fast time digitization

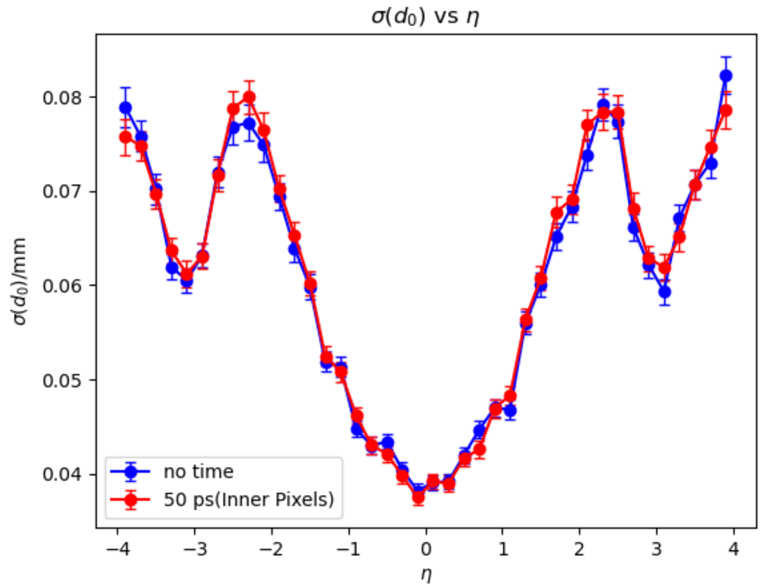
digitized time
stored in RDO

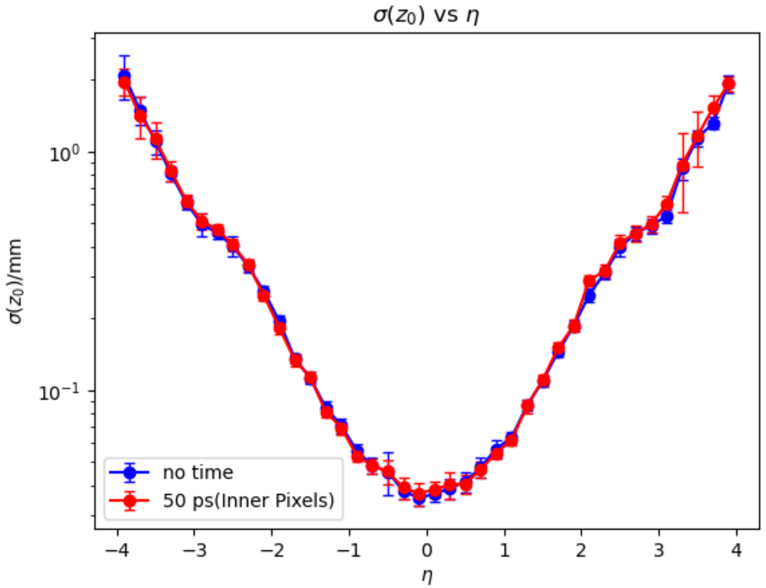
more realistic digitization
(the same method with HGTD)

example 1.732 ns 1.851 ns 00011010…

Purity and track-only impact parameter resolution

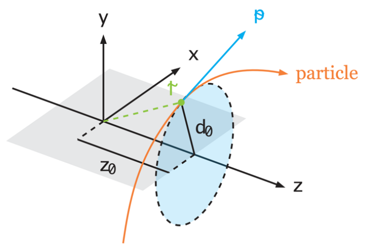




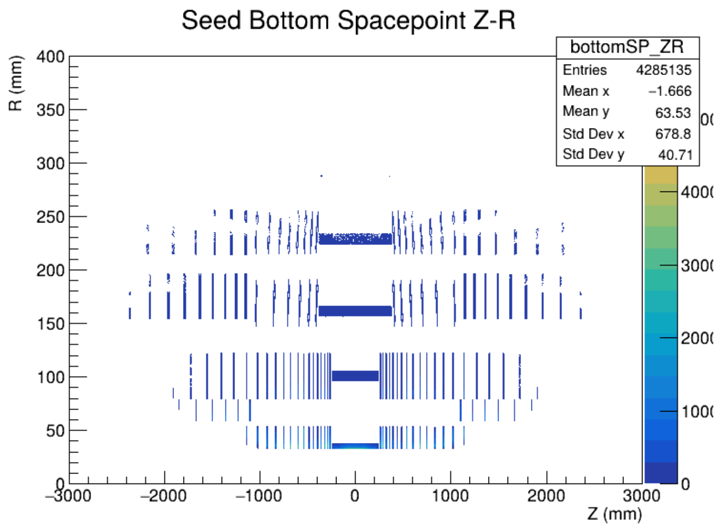


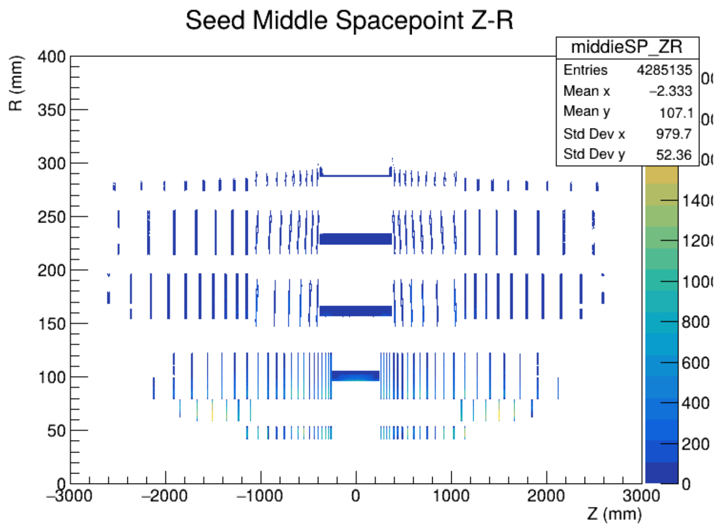
$$\frac{nMajorityHits}{nMeasurements} = \frac{\#\ hits\ that\ belongs\ to\ the\ particle\ with\ the\ highest\ fraction\ of\ hits\ on\ the\ track}{\#\ total\ measuremts\ on\ a\ track}$$

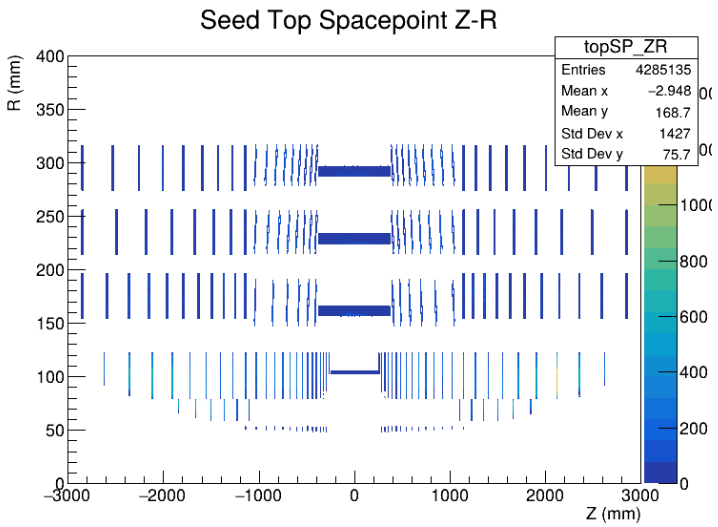
No significant change on the resolution of impact parameters d_0/z_0



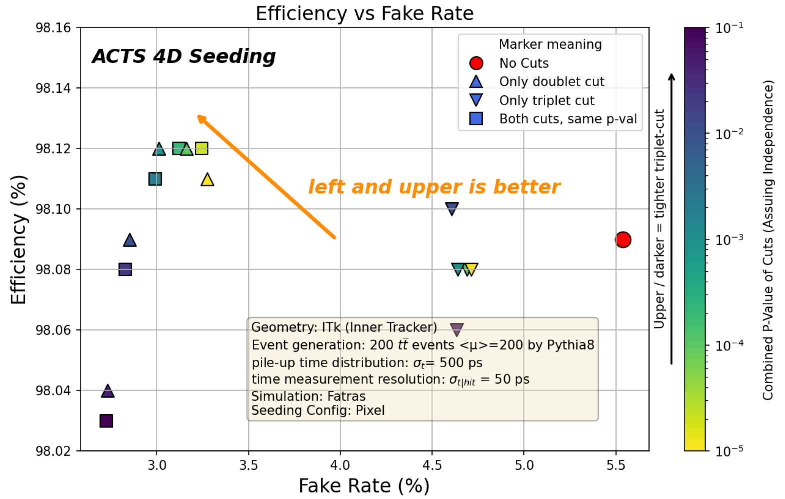
ITk spacepoint distribution

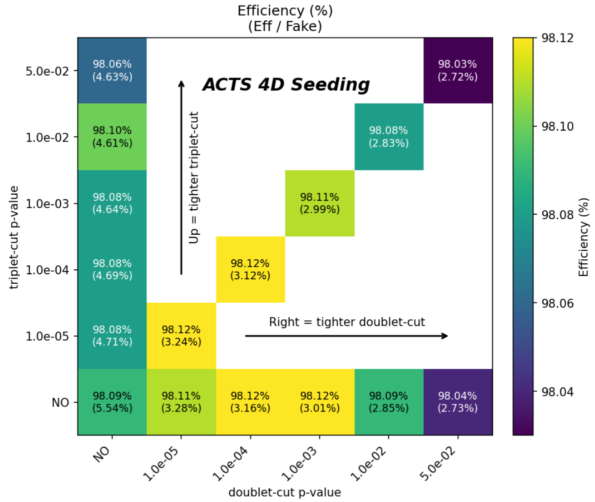


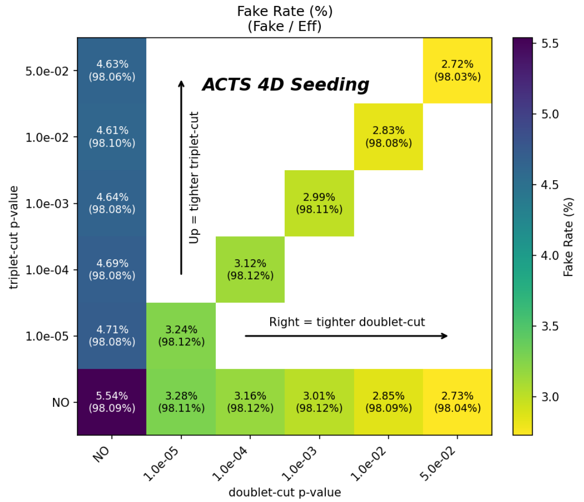


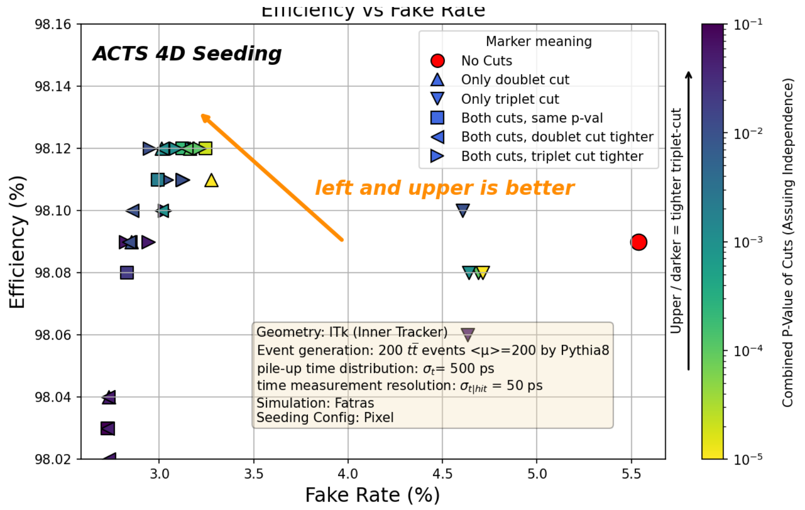


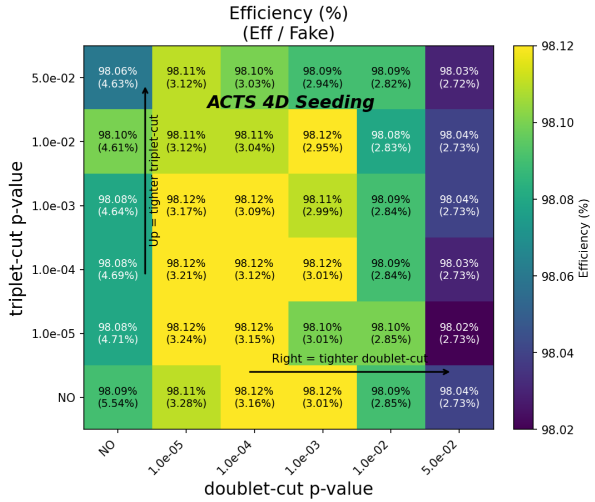
4D seeding performance with ITk

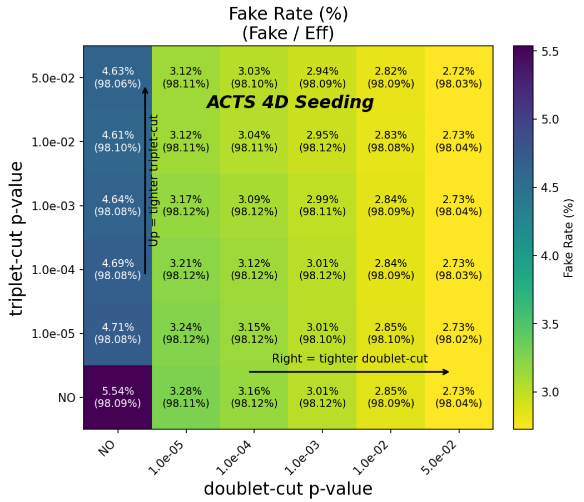












Huffman coding

Sort	Hit-map (bitmask)	Frequency	“6-14” mode bits
1	10101010	6.966%	6
2	01010101	6.418%	6
3	01000000	6.183%	6
4	00000010	5.784%	6
5	00000011	5.175%	6
6	11000000	4.533%	6
7	00000001	3.960%	6
…	…	…	
64	11101010	0.173%	6+8
65	11110100	0.173%	6+8
66	00110101	0.150%	6+8
…	…	…	

Encoding of hit map		Entropy
Bitmask		8
Binary-tree		7.79
Independent address		11.20
Multi-stages Hoffman	4-6-14	7.17
	3-6-14	7.84
	6-14	6.34

Encoding schemes

Huffman Encoding schemes (eg. 6-14 mode):

- For the first 63 high-frequency hit-maps: use 6 bits to encode.
- Since 6 bits correspond to 64 combinations, the last combination will be used as the header for other 193 hit-maps.
- For the other 193 hit-maps: 6 bits header + the original bitmask hit-map (8 bits)

Bitmask: 1 bit for 1 pixel, 1 for hit and 0 for not