TMTT Duplicate Tracks Removal

Luis Ardila*

luis.ardila@stfc.ac.uk

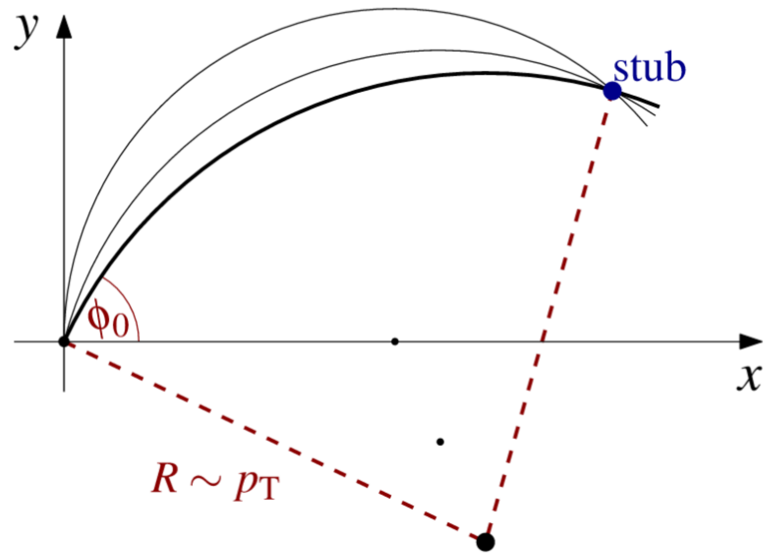
*Supported by the EU FP7-PEOPLE-2012-ITN project nr 317446, INFIERI, "Intelligent Fast Interconnected and Efficient Devices for Frontier Exploitation in Research and Industry”

Outline

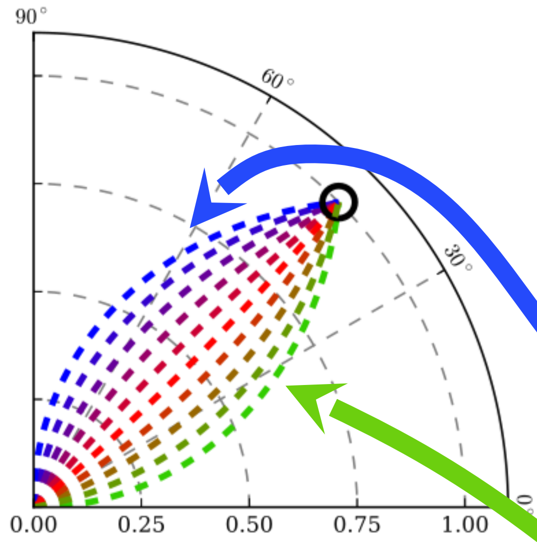
- TMTT Track Removal
    - Duplicate track source
    - Current algorithms to eliminate duplicate tracks
    - Firmware implementation and results
- Evaluation of GPUs for CMS L1 Track Trigger
    - Heterogeneous system performance
- Conclusions

Hough Transform Theory

- search for primary tracks in the r-ϕ plane
- infinite amount of different circles (ϕ_0, R) possible between origin and single measured stub position (r, ϕ)



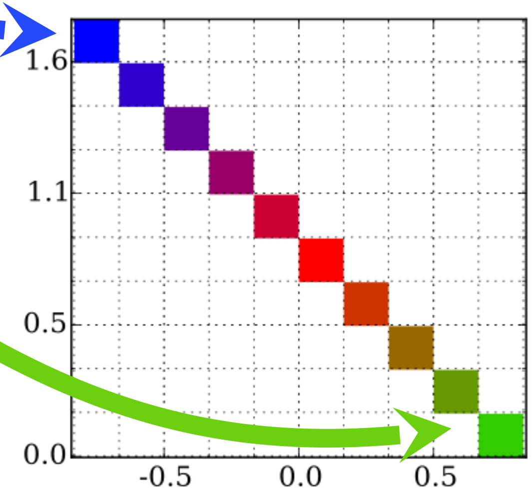
Hough Transformation



Calculate possible

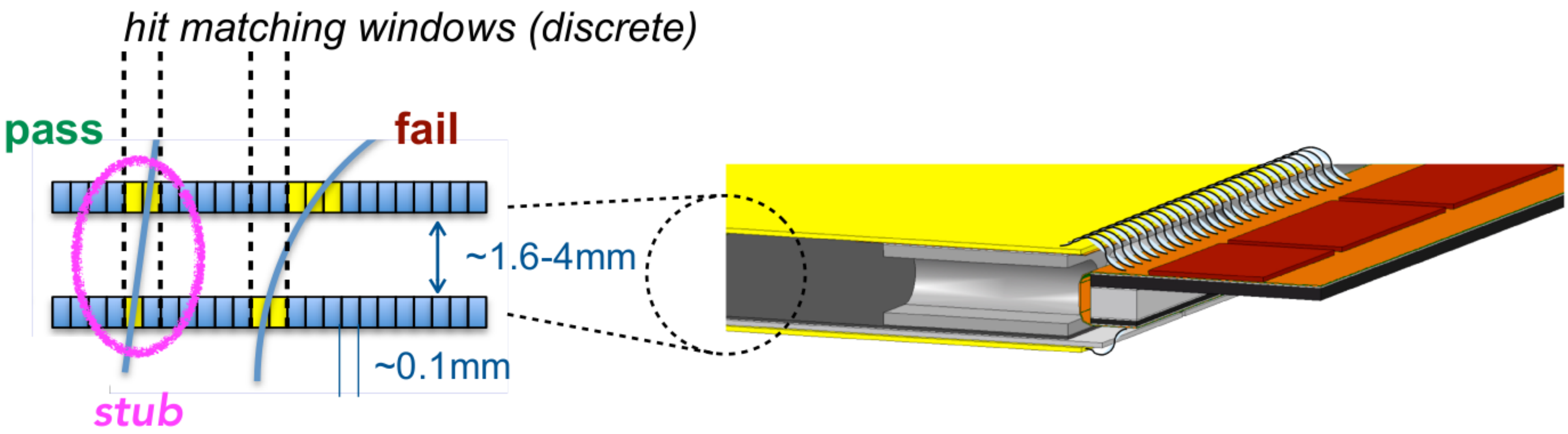
(φ_0, q/pt) pairs for each hit

Make histogram in Hough space



KIT, Institute for Data Processing and Electronics (IPE)
STFC Rutherford Appleton Laboratory

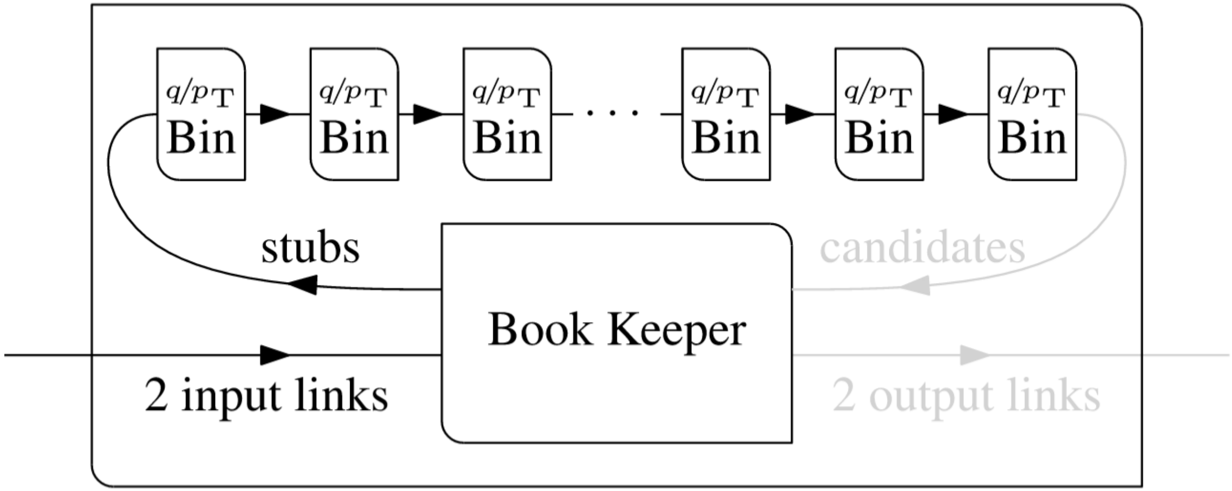
Stub Building



- Applies momentum cut to hits
- Delivers estimate on track bend
- Drastically decreases number of hits by a factor of 100

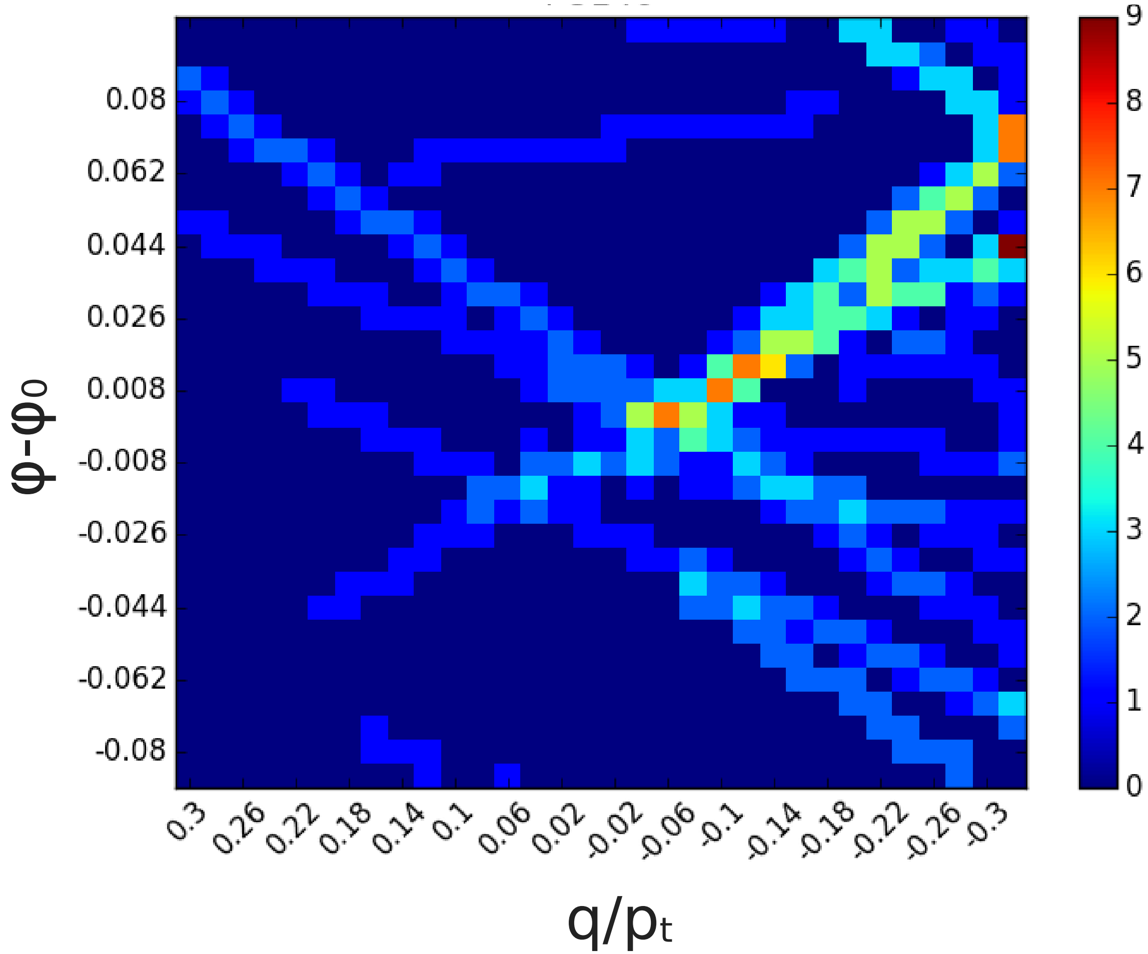
Hough Transform in Firmware

- Hough space is implemented as a pipeline, it processes one stub per clock cycle (240 MHz)
- Stubs can be filed inside multiple bins → creating possible duplicate tracks



- Only the bins containing stubs in 5 or more layers are considered to be possible track

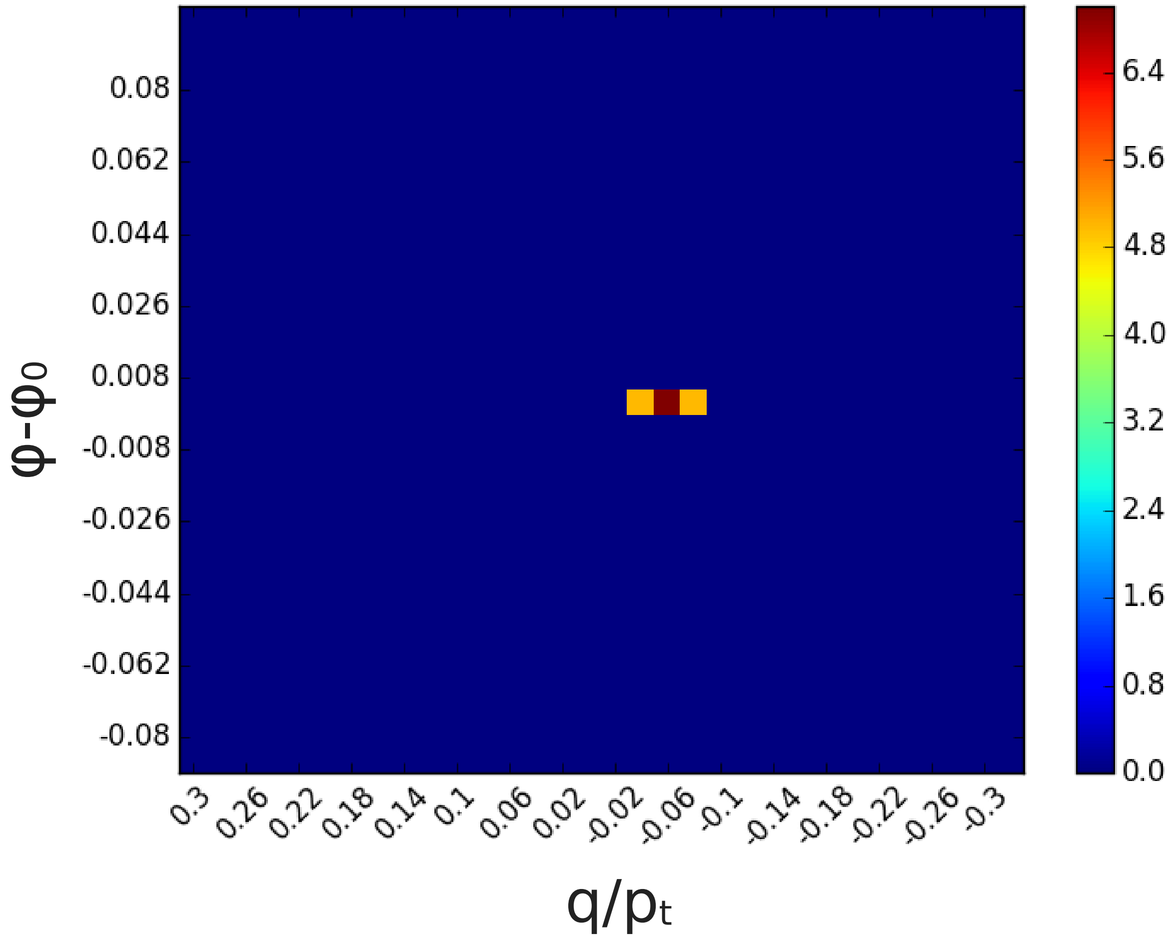
Clustering points are track candidates



TTBar event - PU 140

KIT, Institute for Data Processing and Electronics (IPE)
STFC Rutherford Appleton Laboratory

Clustering points are track candidates

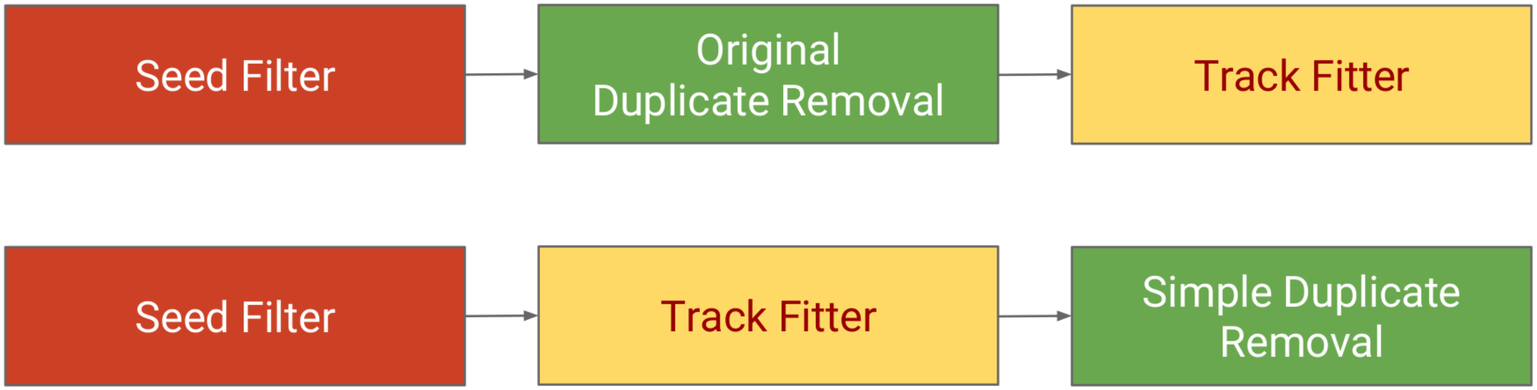


TTBar event - PU 140

Duplicate Removal Options

Two possible duplicate removal algorithms can be run in different parts of the chain

- Original Duplicate Removal: placed after the Seed Filter
- Simple Duplicate Removal: placed at the end of the chain



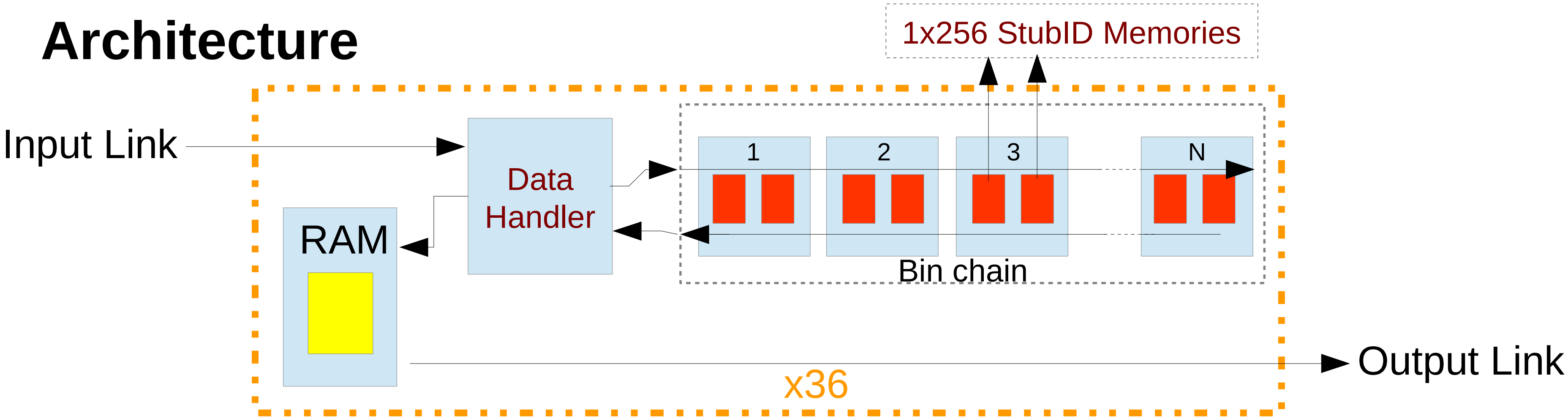
“Original” Duplicate Removal

- Compares all tracks in a given event, two tracks at a time, if they have stubs in common in more than 5 tracker layers then they are duplicates

- The **track with more layers is preserved**

- If both tracks have the same number of layers, then the **track with higher pT is preserved**

“Original” Duplicate Removal Firmware Architecture



- Design is composed by 35 identical “bins”, each containing two 256 bit memories
- When track “j” arrives, it progresses through each “Bin” (1 to j - 1) in turn, before finally being stored in “Bin j”.
    - A bit is set in the “Bin j” RAM at address corresponding to the stub identifier (integer in range 1-210) of each stub on track “j”
- As each new track is processed & passes through Bins already containing tracks, it uses the RAM in each Bin to check if the new track shares stubs in common with the stored track

KIT, Institute for Data Processing and Electronics (IPE)
STFC Rutherford Appleton Laboratory

Problems of the “Original” Duplicate Removal Algorithm

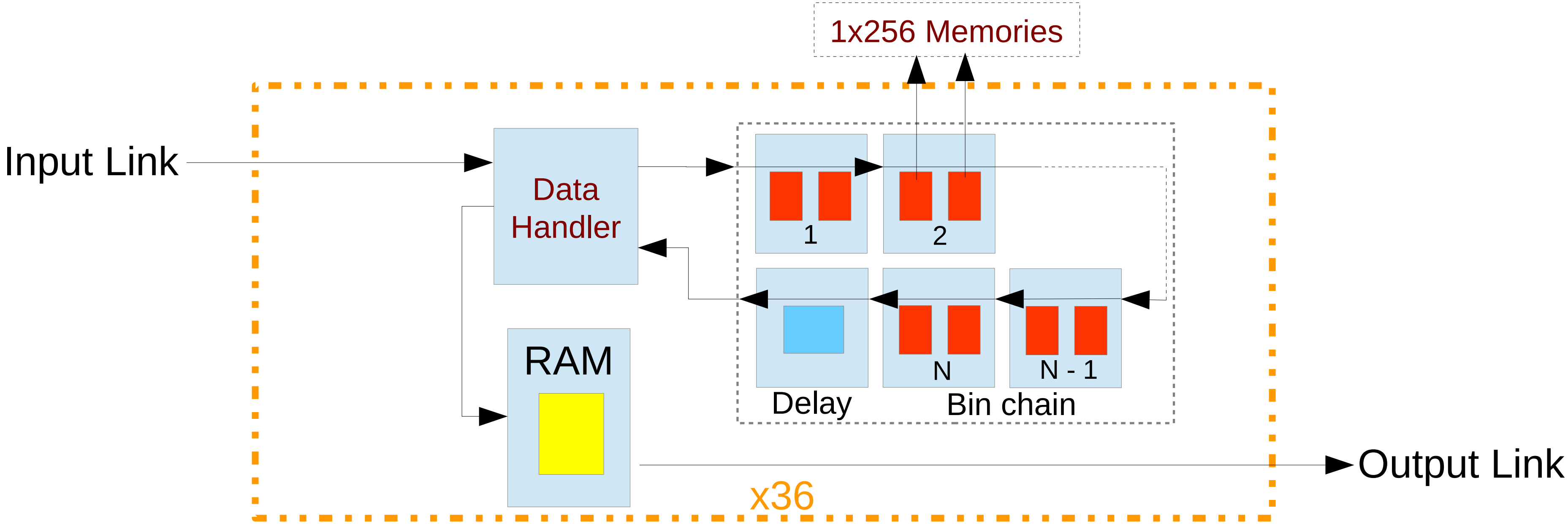
- High Latency: in the order of one time multiplex period ~900 ns
- Requires all data from a given sector in a single pair of links
- Uses about half of the available resources in an MP7 board

“Original” V2.0
Duplicate Removal

- Variant of “Original” Algorithm → Compares a limited range of tracks in an event, two at a time, if they have stubs in common
  - Profiting from ~pT ordering of input tracks

- If there are stubs in common in more than 5 tracker layers then the **track with more layers is preserved**
- If both tracks have the same number of layers, then the **track with higher pT is preserved**

“Original” V2.0 Duplicate Removal Firmware



- Tracks are shifted and stored in empty bins, when the Track of bin j arrives back in the Data Handler it is evaluated if bin j is valid or not

- Fixed latency determined by the bin number (35) and the chosen delay (currently 16) = 51 clock cycles @ 240 MHz (~212 ns)

Resources utilization: 36 sectors + mp7 infrastructure

“Original”:

Latency : ~904 ns (217 clk cycles)

Resource	Estimation	Available	Utilization %
LUT	198382	433200	45.79
LUTRAM	11769	174200	6.76
FF	218172	866400	25.18
BRAM	361	1470	24.56

“Original” V2.0:

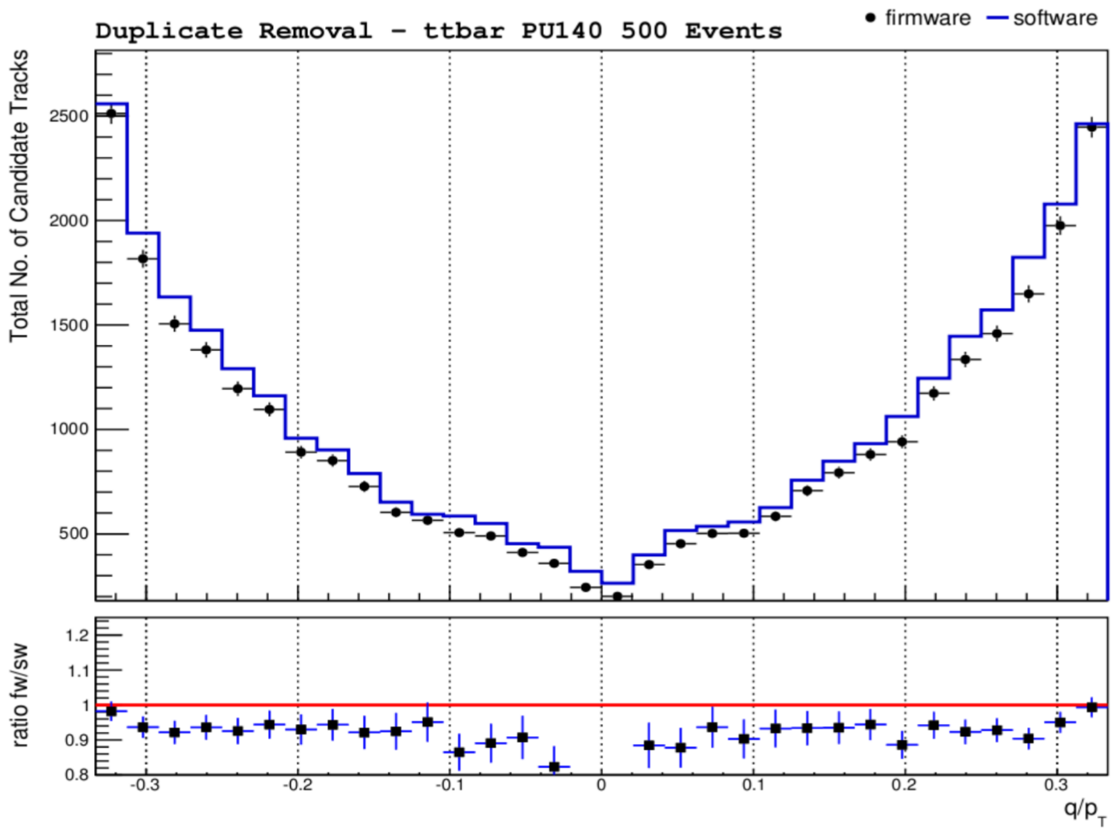
Latency : ~212 ns (51 clk cycles)

Resource	Estimation	Available	Utilization %
LUT	210791	433200	48.66
LUTRAM	14180	174200	8.14
FF	210468	866400	24.29
BRAM	453	1470	30.82

Results

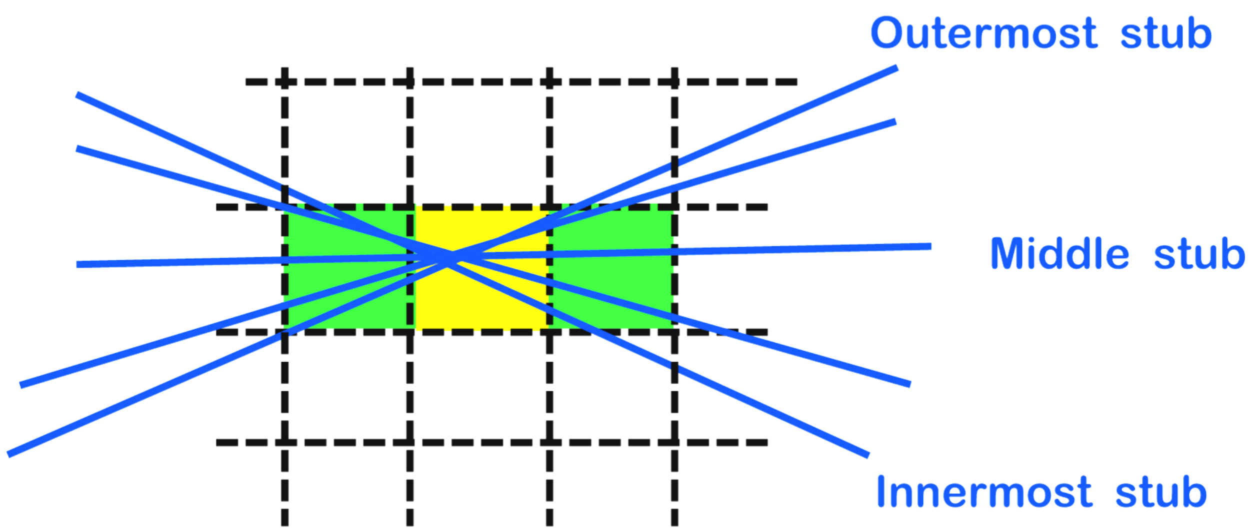
Fair agreement between tracks produced by duplicate track removal running on MP7 hardware vs. predictions from analysis software

◆ Inconsistencies caused by using older version of algorithm in firmware than in software.



"Simple" Track Duplicate Algorithm

- In the Hough transform shown, the 5 stubs (blue lines) from a single particle produce 3 track candidates in the green & yellow HT cells
- These three tracks contain the same stubs, so when they are fitted, they all yield identical fitted helix parameters
- These fitted helix parameters should correspond to the yellow cell, where the lines intersect. (Although resolution effects may change this ...)



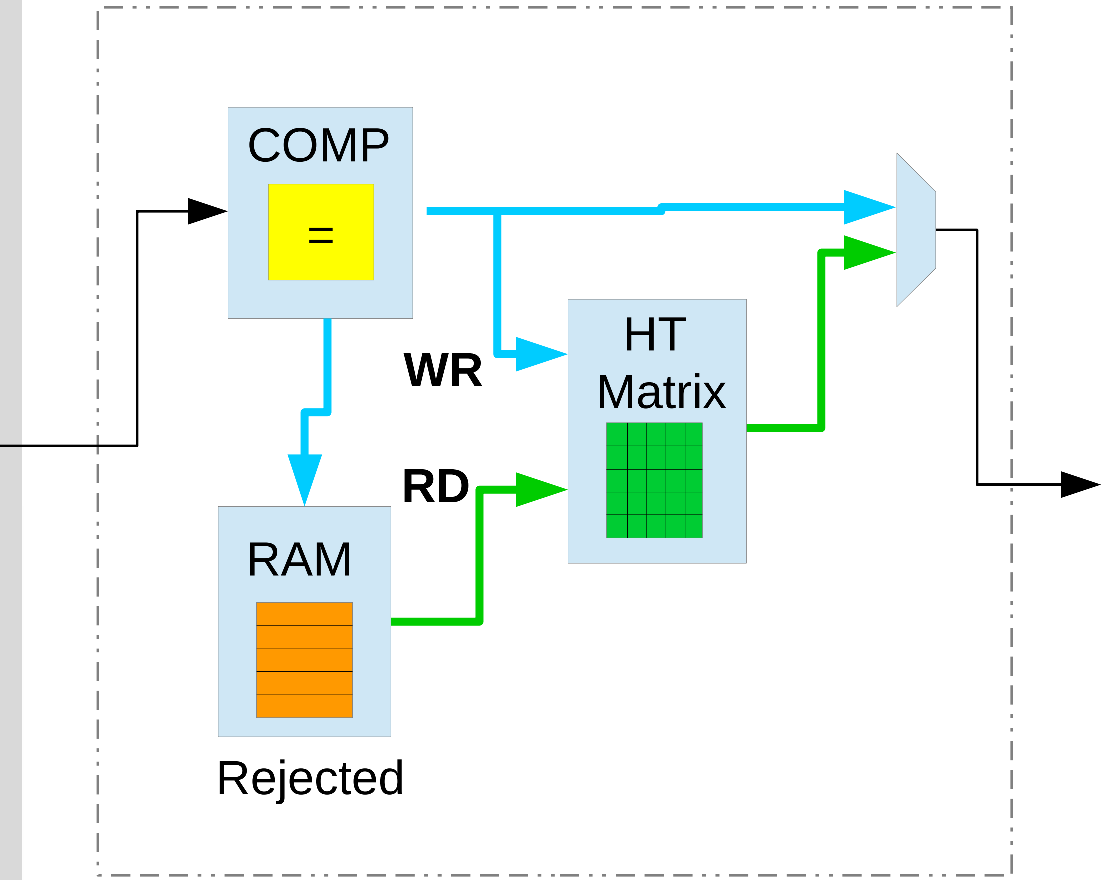
“Simple” Duplicate Removal Algorithm

Algorithm: after fit, kill any tracks if their fitted helix parameters don’t correspond to the same HT cell as the HT originally found the track in. (i.e. In example, kill green cells & keep yellow one)

N.B. This new algorithm finds duplicates by looking at individual tracks. We no longer need to compare pairs of tracks to see if they are the same as each other!

Subtlety: The above algorithm loses a few percent efficiency due to resolution effects. To recover this, we have a 2^{nd} pass through the rejected tracks. Any whose fitted helix parameters do not correspond to an HT cell of an accepted track from the 1^{st} pass are rescued.

“Simple” Duplicate Track Removal Firmware Architecture



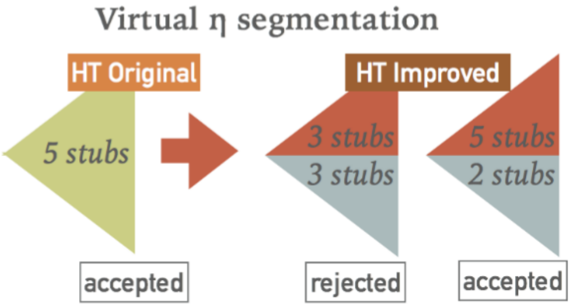
- During 1st pass (blue arrows), tracks whose fitted helix parameters correspond to the same HT cell that the HT originally found the track in (“COMP” block) are transmitted to output, and cell is stored in “HT Matrix”.
- During 2nd pass (green arrow) tracks rejected by 1st pass (“Rejected”) are rescued if fitted helix parameters don’t correspond to HT cell stored in “HT Matrix”

“Simple” Duplicate Removal Remarks

- It runs after the track fit
- Low Resource Utilization ~15% MP7 Virtex7 690 FPGA
- Low Latency ~100 ns
- The algorithm is based on an understanding of how duplicate tracks form within the Hough transform
- No need to compare pairs of tracks

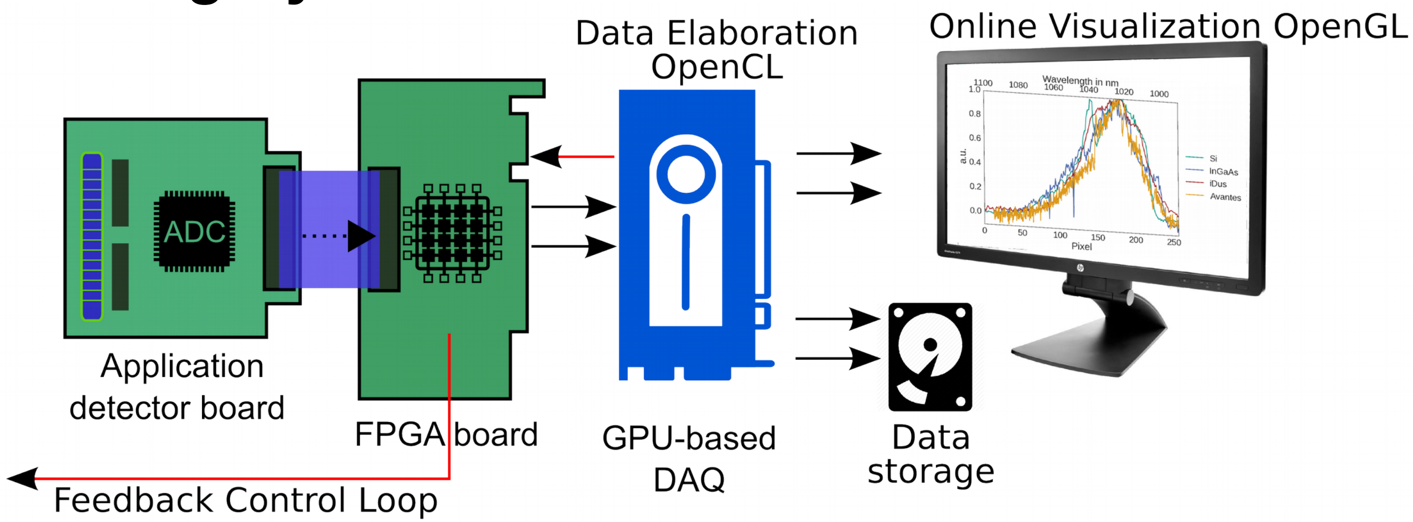
Improved HT

- HT configured for new geometric segmentation (18 x 2)
- Granularity per array increased (32 → 64 rows in φ, no change in q/pT)
  - More pages in block RAM, but same utilization
- Addition of virtual segmentation in η, per bin, to reduce output rate

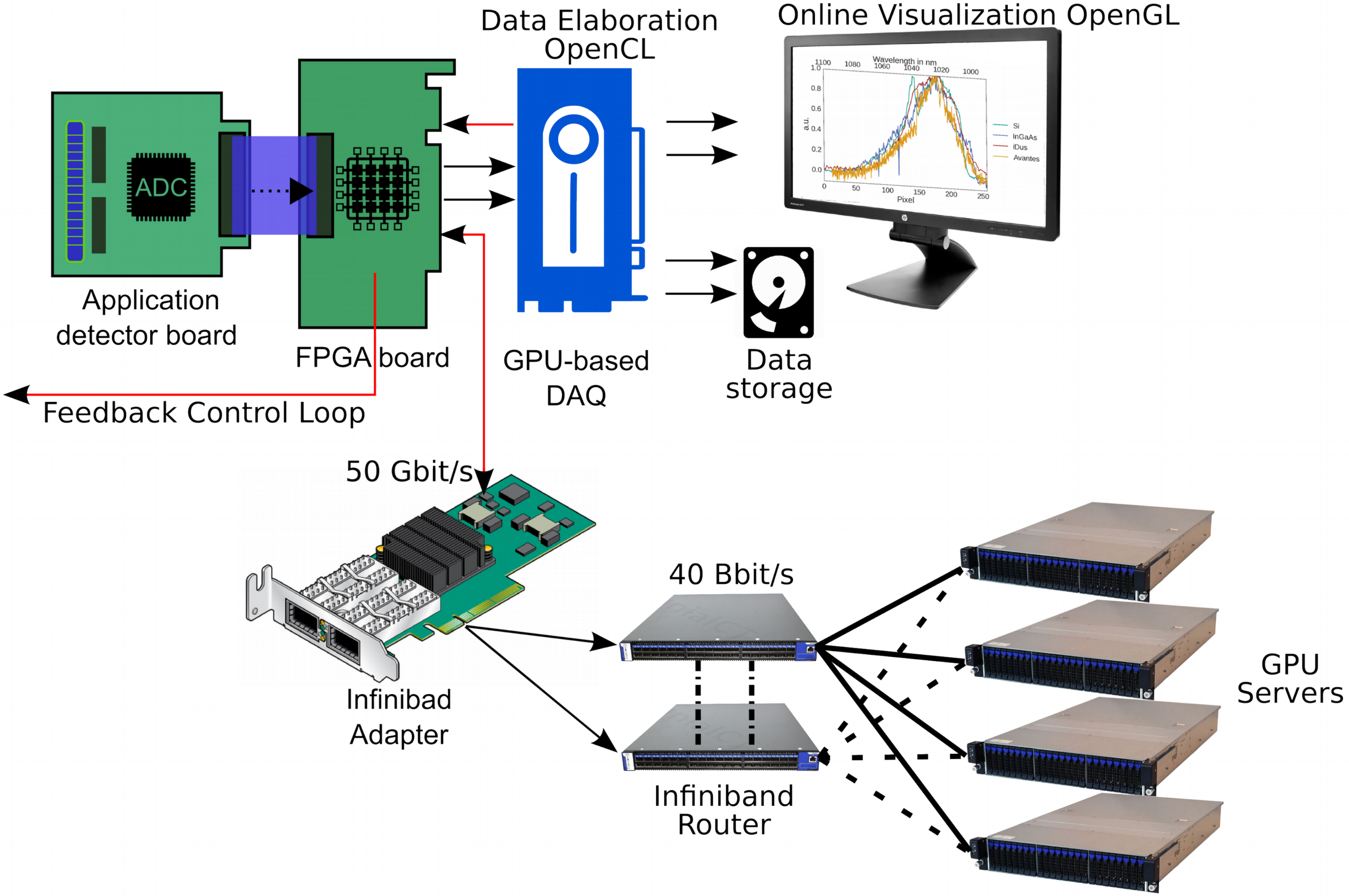


segmentation (η, ϕ)	η sub sectors	HT array (q/p_T, ϕ_{58})	# output candidates	algorithmic efficiency	purity
9 x 4	No	32 x 32	766	97.0%	28%
9 x 4	Yes	32 x 32	352	97.1%	27%
18 x 2	Yes	32 x 64	255	97.3%	43%

Heterogeneous Data Processing System

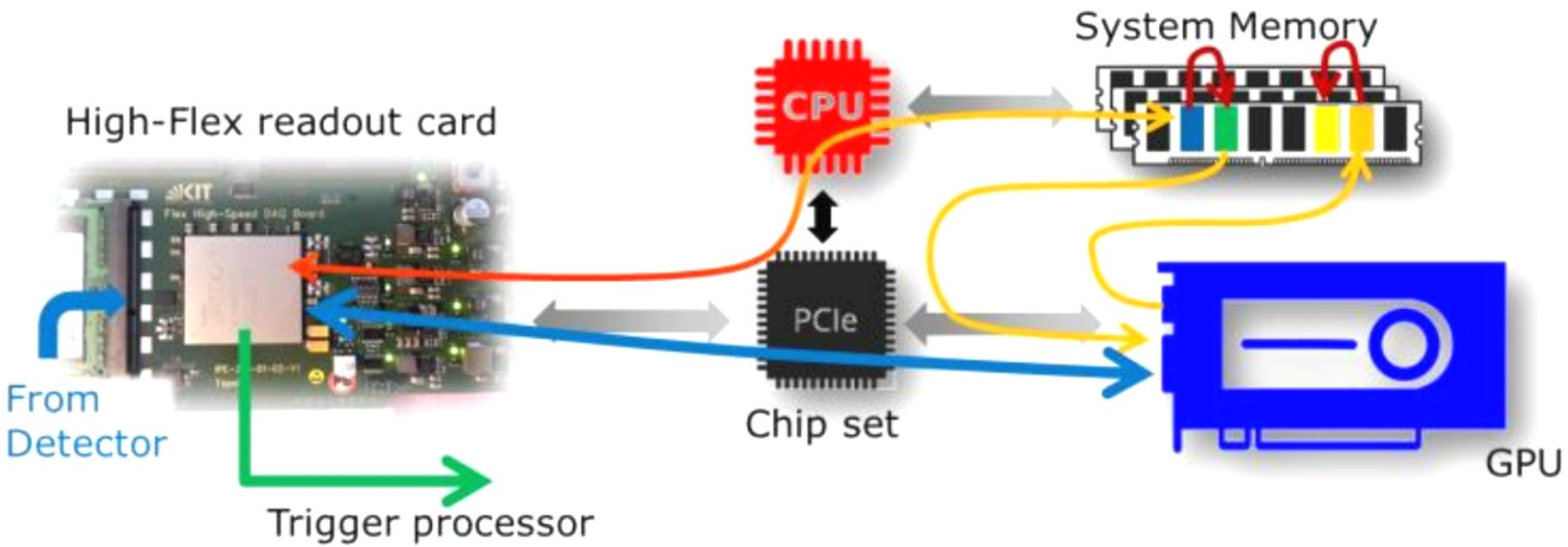


Heterogeneous Data Processing System



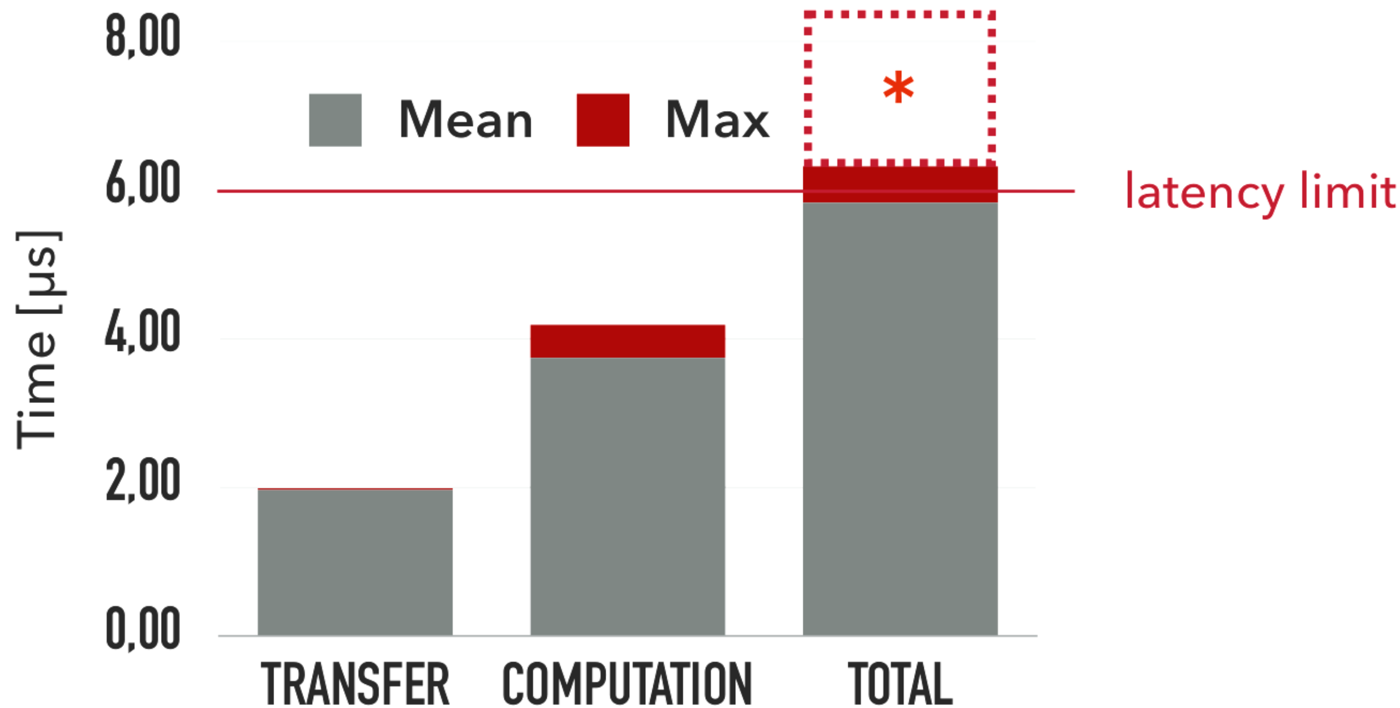
KIT, Institute for Data Processing and Electronics (IPE)
STFC Rutherford Appleton Laboratory

“GPUDirect” (NVIDIA) “DirectGMA” (AMD) Technologies



DMA benchmark - HT

➤ Read/Uncompress data ➤ Compute ➤ Poll

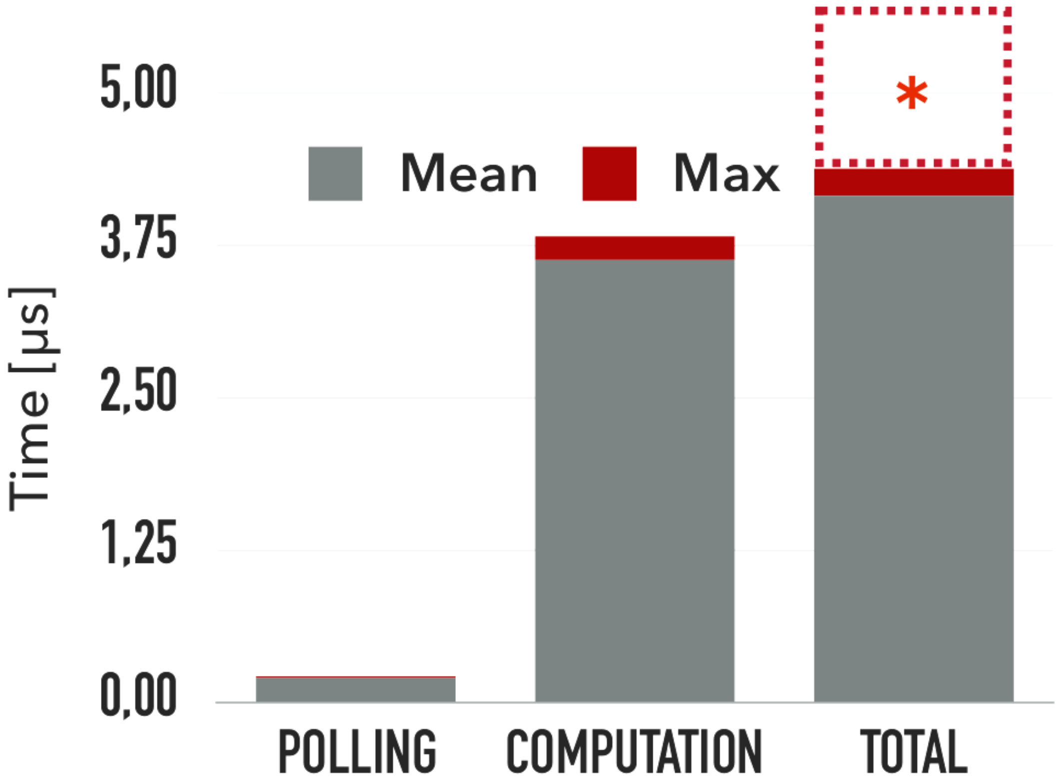


160 stubs, 1 sector

* ···· estimated response time

DMA benchmark: Interleaved HT

➤ (poll) Read/Uncompress data ➤ Ask for data ➤ Compute

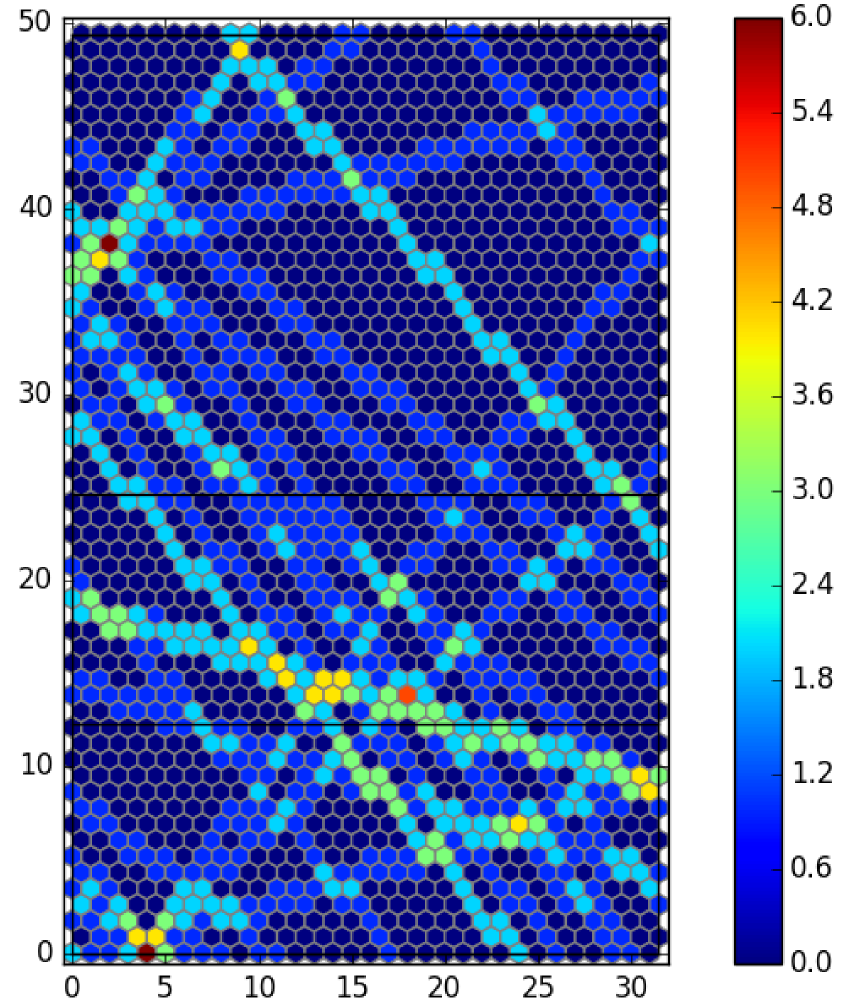


* ···· estimated response time

KIT, Institute for Data Processing and Electronics (IPE)
STFC Rutherford Appleton Laboratory

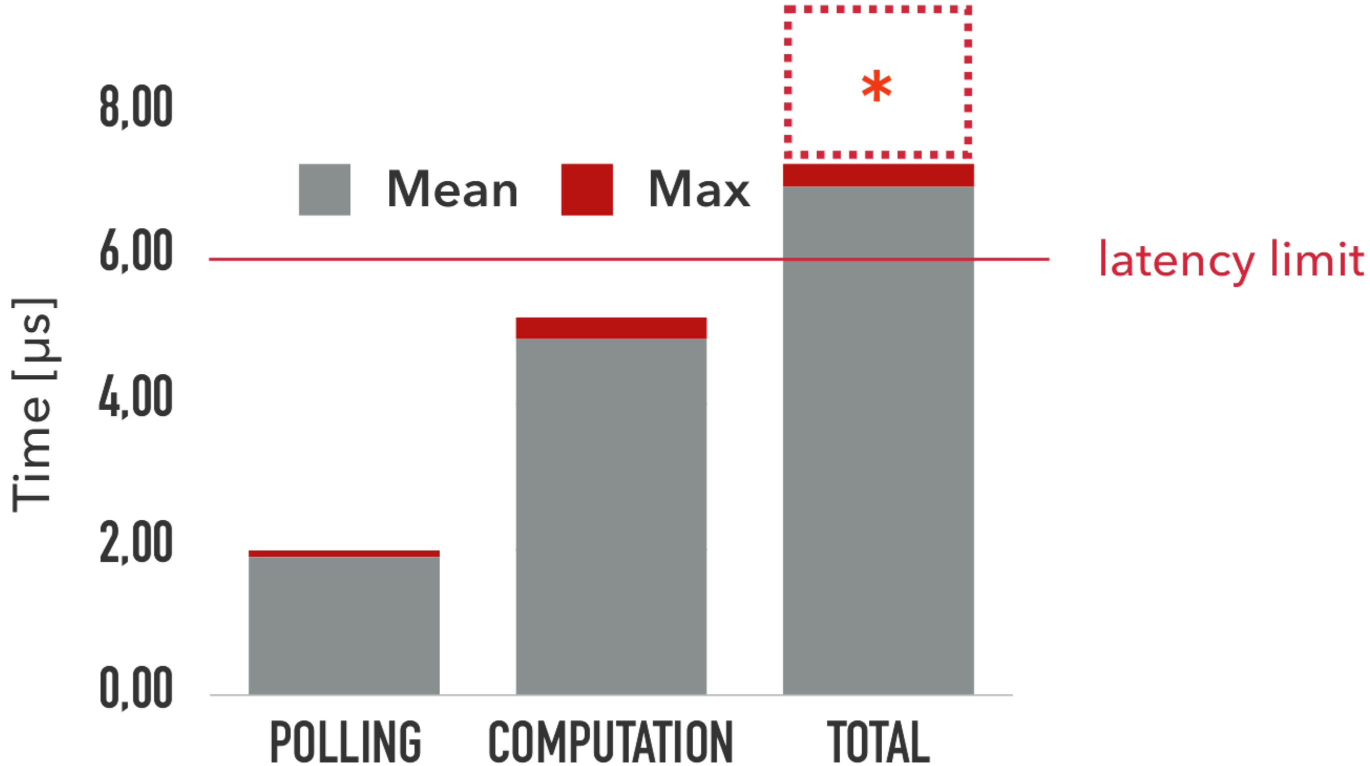
Hexagon Hough space

- Hexagonal bins in hough space
- Suppresses fake candidates
- Runtime comparable
- only 1 possible bin per row
- less algorithmic branching



DMA benchmark: Hexagonal HT

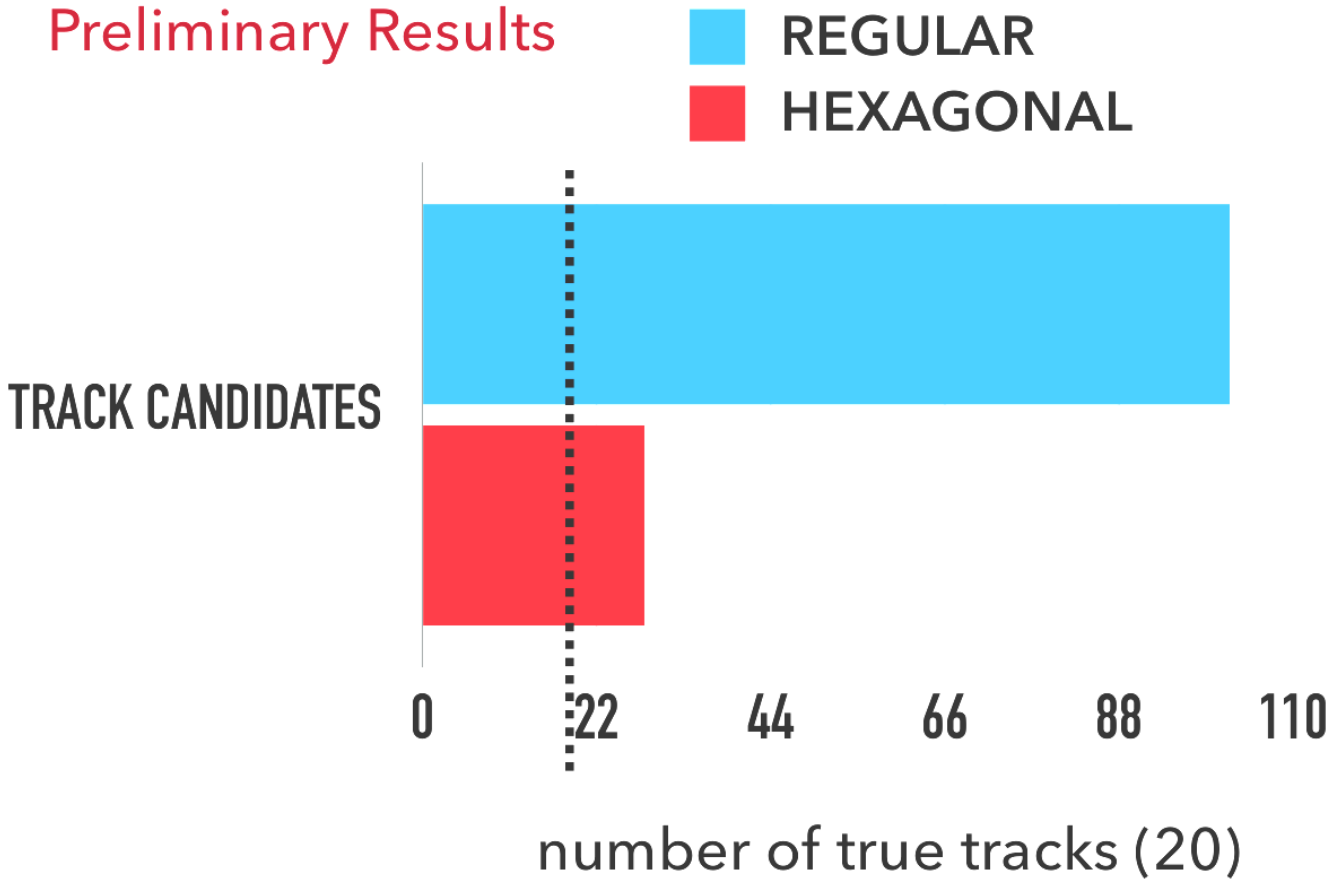
➤ (poll) Read/Uncompress data ➤ Ask for data ➤ Compute



* estimated response time

Comparison of fake rate Regular vs Hexagon HT

Preliminary Results



Results for TTBar Dataset PU140, whole detector, 1 event

Conclusions

- The “Original” track duplicate removal algorithm is implemented in hardware and we are tuning the agreement between software and firmware to validate the design
- The “Simple” track duplicate removal algorithm is being developed and integrated with both of the fitter stages,
- Changes in the segmentation of the detector and improvements to the Hough Transform allows us to increase the granularity, increment the purity of the produced tracks and lower the output rate of candidates therefore producing less duplicate tracks

- Real-time performance of heterogeneous systems is on track to be on par in the following years compared to fully deterministic FPGA designs with the advantage of algorithmic flexibility and faster developing times