

# DHH Clustering

**Igor Konorov**

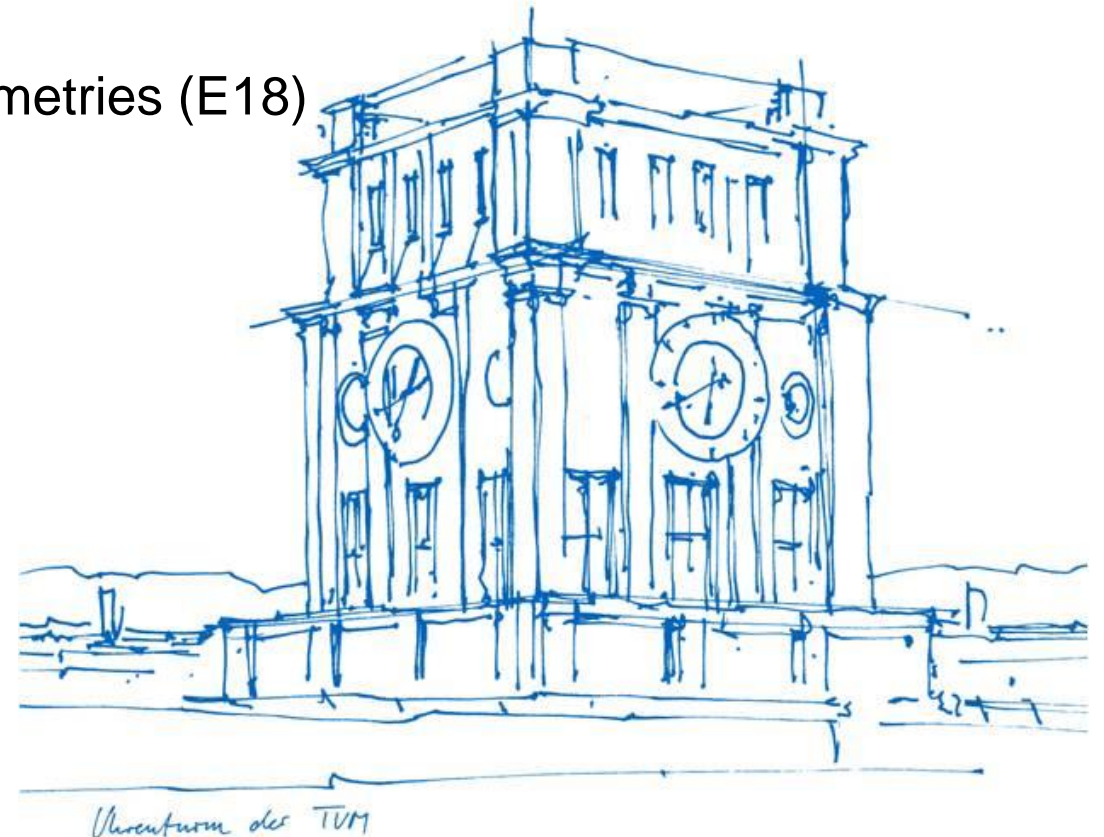
Institute for Hadronic Structure and Fundamental Symmetries (E18)

TUM Department of Physics

Technical University of Munich

FSP Workshop : Slow pion tracking

8-12-th November 2021



# Clustering algorithm for FPGA

## Simple requirements:

Merging direct neighbors :

or



Pixel array 768x250

Real time processing

## Required performance

- 2 % occupancy => 4k hits
- 30 kHz => 33us/event => 8.25 ns/hit or 125 Mhit/s

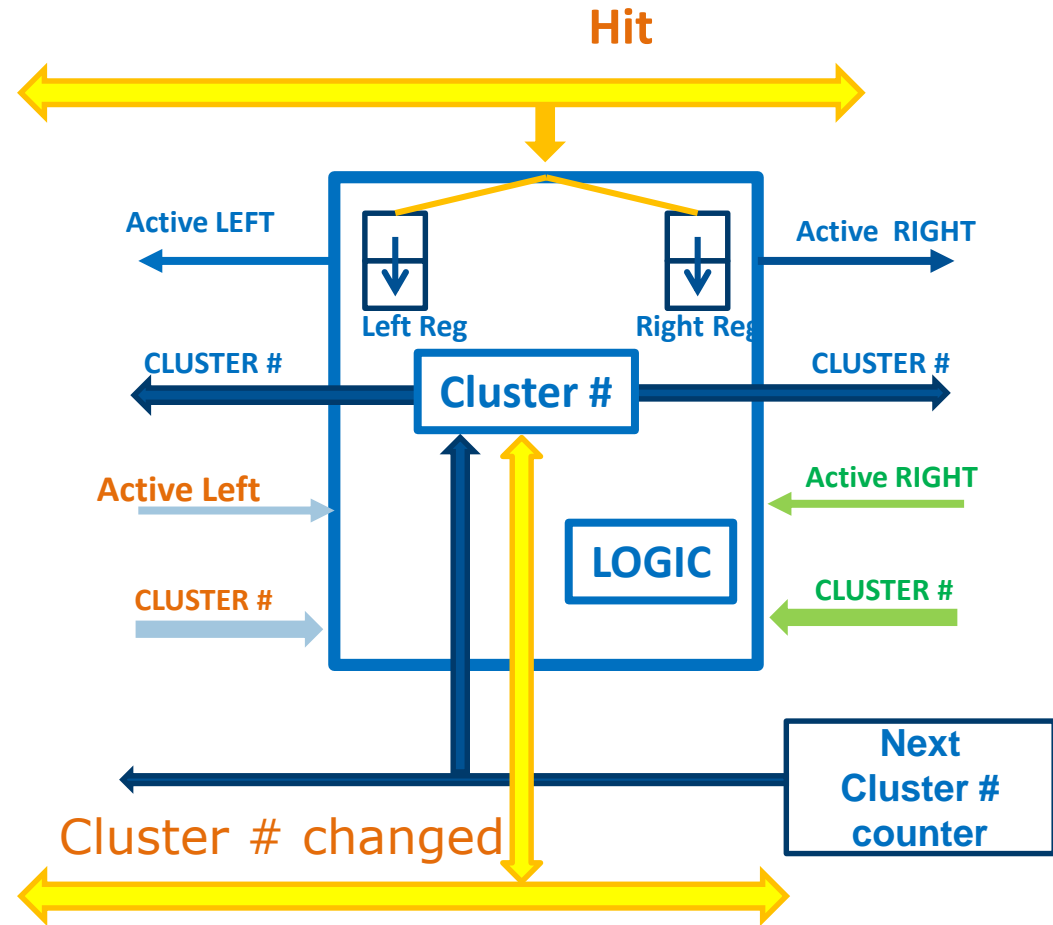
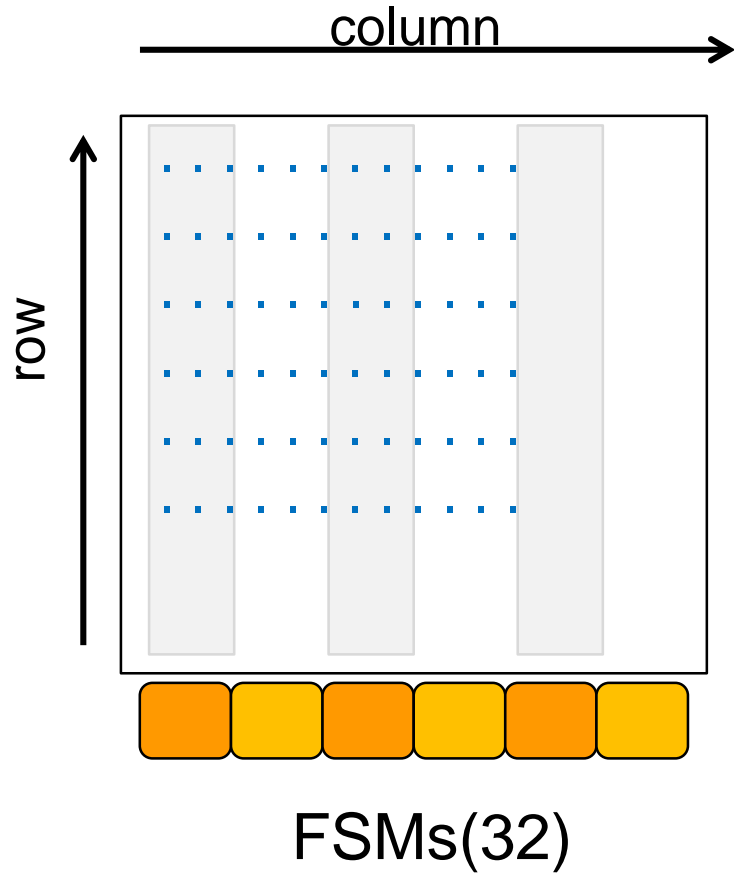
## Algorithm takes advantage of DHP readout features :

- sequential data transmission
- Ordered hits readout sequence, almost row wise:
  - data mixed within 4 consecutive rows
- no latency requirements

## Clustering algorithm features:

- Hit information is analyzed once and cluster number assigned
- Following processing steps shuffle hits using cluster number information
- Clustering algorithm reconstructs any cluster shape within half ladder
- Pipeline design – real time operation

# Clustering FSMs



What **FSMs** do?

- Each FSM responsible for hits of two columns
- Process one hit in one clock cycle
- Evaluate **hit cluster number**
- Write hit together with cluster number to hit memory
- Store cluster number in **cluster memory**
- **When two clusters touch each other the lowest cluster number is taken over**

## **FSM behavior**

1. FSM is not active, hit arrives
2. FSM is active, no hit
3. FSM is active and new hit arrives
4. FSM is active, there was no hit belonging to any of these two rows within this column and current hit is a first belonging to new column

```

. . . . .
. . . X . . .
. . . . .
. . . X . . .
. . . . .
. X X . . . .
. . . . .
. . X . . X .
. . . X . X .
. . . . X . .
    
```

Next cluster counter

1
---

Cluster memory

Addr.	Value
1	-
2	-
3	-
4	-
5	-

```

. . . . .
. . . 1 . . .
. . . X . . .
. . . . .
. X X . . . .
. . . . .
. . X . . X .
. . . X . X .
. . . . X . .

```

Next cluster counter

2
---

Cluster memory

Addr.	Value
1	1
2	-
3	-
4	-
5	-

```

. . . . .
. . . 1 . . .
. . . . .
. . . 2 . . .
. . . . .
. X X . . . .
. . . . .
. . X . . X .
. . . X . X .
. . . . X . .
    
```

Next cluster counter

3
---

Cluster memory

Addr.	Value
1	1
2	2
3	-
4	-
5	-



# Examples of FSMs actions 1

```

. . . . .
. . . 1 . . .
. . . . .
. . . 2 . . .
. . . . .
. 3 X . . . .
. . . . .
. . X . . X .
. . . X . X .
. . . . X . .
    
```

Next cluster counter

4
---

Cluster memory

Addr.	Value
1	1
2	2
3	3
<b>4</b>	-
<b>5</b>	-

# Examples of FSMs actions 1

```

. . . . .
. . . 1 . . .
. . . . .
. . . 2 . . .
. . . . .
. 3 3 . . .
. . . . .
. . X . . X .
. . . X . X .
. . . . X . .

```

Next cluster counter

4
---

Cluster memory

Addr.	Value
1	1
2	2
3	3
<b>4</b>	-
<b>5</b>	-

# Examples of FSMs actions 1

```

. . . . .
. . . 1 . . .
. . . . .
. . . 2 . . .
. . . . .
. 3 3 . . .
. . . . .
. . 4 . . X .
. . . X . X .
. . . . X . .
    
```

Next cluster counter

5
---

Cluster memory

Addr.	Value
1	1
2	2
3	3
4	4
5	-

```

. . . . .
. . . 1 . . .
. . . . .
. . . 2 . . .
. . . . .
. 3 3 . . .
. . . . .
. . 4 . . 5 .
. . . X . X .
. . . . X . .
    
```

Next cluster counter

6
---

Cluster memory

Addr.	Value
1	1
2	2
3	3
<b>4</b>	4
<b>5</b>	5

# Examples of FSMs actions 1

```

. . . . .
. . . 1 . . .
. . . . .
. . . 2 . . .
. . . . .
. 3 3 . . .
. . . . .
. . 4 . . 5 .
. . . 4 . 5 .
. . . . X . .
    
```

Next cluster counter

6
---

Cluster memory

Addr.	Value
1	1
2	2
3	3
<b>4</b>	4
<b>5</b>	5

# Examples of FSMs actions 1

```

. . . . .
. . . 1 . . .
. . . . .
. . . 2 . . .
. . . . .
. 3 3 . . .
. . . . .
. . 4 . . 5 .
. . . 4 . 5 .
. . . . 4 . .
    
```

Next cluster counter

6
---

Cluster memory

Addr.	Value
1	1
2	2
3	3
<b>4</b>	4
<b>5</b>	4

# Examples of FSMs actions 1

```

. . . . .
. . . 1 . . .
. . . . .
. . . 2 . . .
. . . . .
. 3 3 . . .
. . . . .
. . 4 . . 4 .
. . . 4 . 4 .
. . . . 4 . .
    
```

Next cluster counter

6
---

Cluster memory

Addr.	Value
1	1
2	2
3	3
4	4
5	4

Cluster numbers are updated using cluster memory during hit readout.

```

. . . . .
. . . 1 . . .
. . . . .
. . . 2 . . .
. . . . .
. 3 3 . . . .
. . . . .
. . 4 . . 5 .
. . . 4 . 5 .
. . . . X . .
    
```

Next cluster counter

6
---

Cluster memory

Addr.	Value
1	1
2	2
3	3
<b>4</b>	4
<b>5</b>	5



```

. X . X . X . X . X . X . X . X . X . . . . .
. X . X . X . X . X . X . X . X X . . . . .
. X . X . X . X . X . X . X X . . . . .
. X . X . X . X . X . X X . . . . .
. X . X . X . X . X X . . . . .
. X . X . X X . . . . .
. X . X X . . . . .
. . X . . . . .

```

Next cluster counter

1
---

Cluster memory

Addr.	Value
1	-
2	-
3	-
<b>4</b>	-
<b>5</b>	-
<b>6</b>	-
<b>7</b>	-
<b>8</b>	-
<b>9</b>	-

.	1	.	2	.	3	.	4	.	5	.	6	.	7	.	8	.	9	.	.	.	.	.
.	1	.	2	.	3	.	4	.	5	.	6	.	7	.	8	8	.	.	.	.	.	.
.	1	.	2	.	3	.	4	.	5	.	6	.	7	7	.	.	.	.	.	.	.	.
.	1	.	2	.	3	.	4	.	5	.	6	6	.	.	.	.	.	.	.	.	.	.
.	1	.	2	.	3	.	4	.	5	5	.	.	.	.	.	.	.	.	.	.	.	.
.	1	.	2	.	3	.	4	4	.	.	.	.	.	.	.	.	.	.	.	.	.	.
.	1	.	2	.	3	3	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
.	1	.	2	2	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	1	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.

Cluster memory content has to be updated to map all cluster numbers to smallest number

Next cluster counter

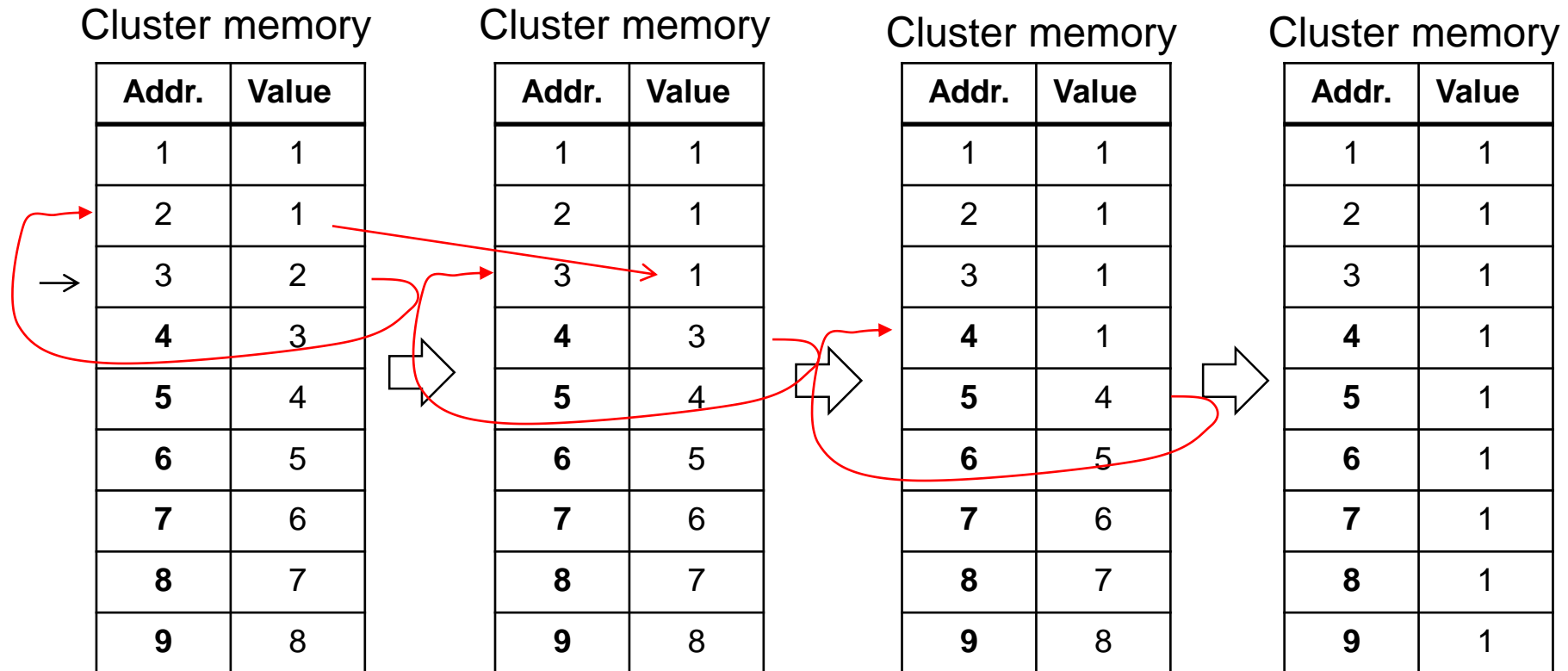
10
----

Cluster memory

Addr.	Value
1	1
2	1
3	2
4	3
5	4
6	5
7	6
8	7
9	8

# TUM Cluster memory update

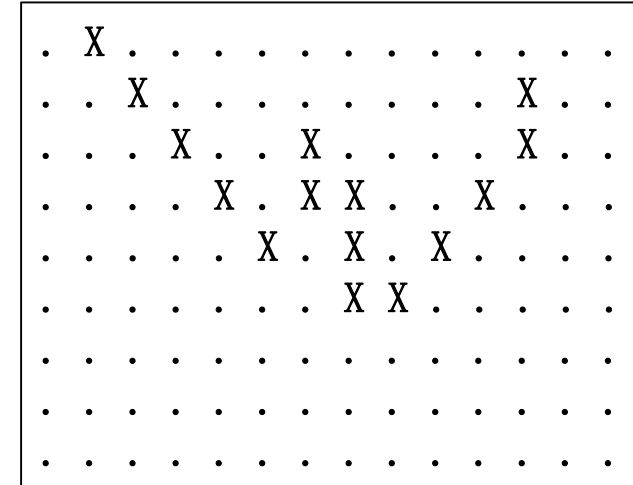
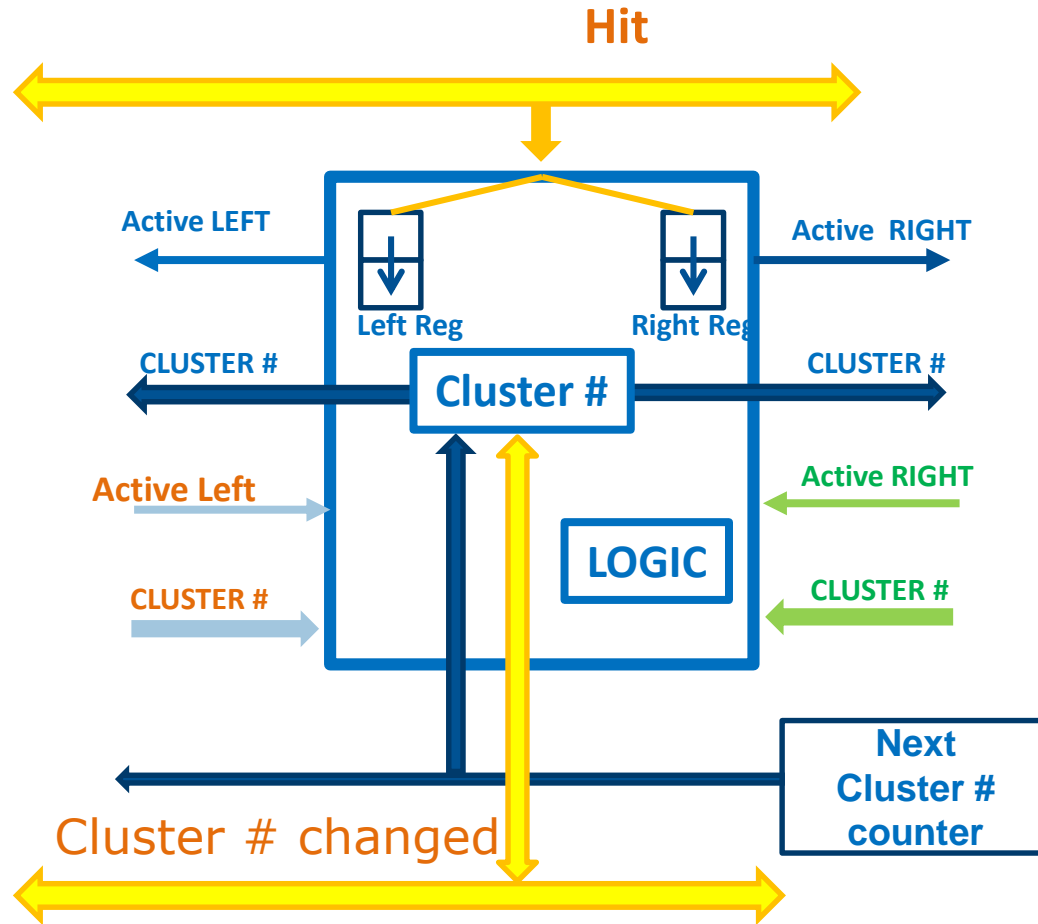
Update starts from cluster #3 to maximum cluster #  
Maximum look down length is 1 clusters



One update takes 3 clock cycles

Using DP memory allows to reach 2 clock cycles per update

# Broadcast Cluster Number change (1)

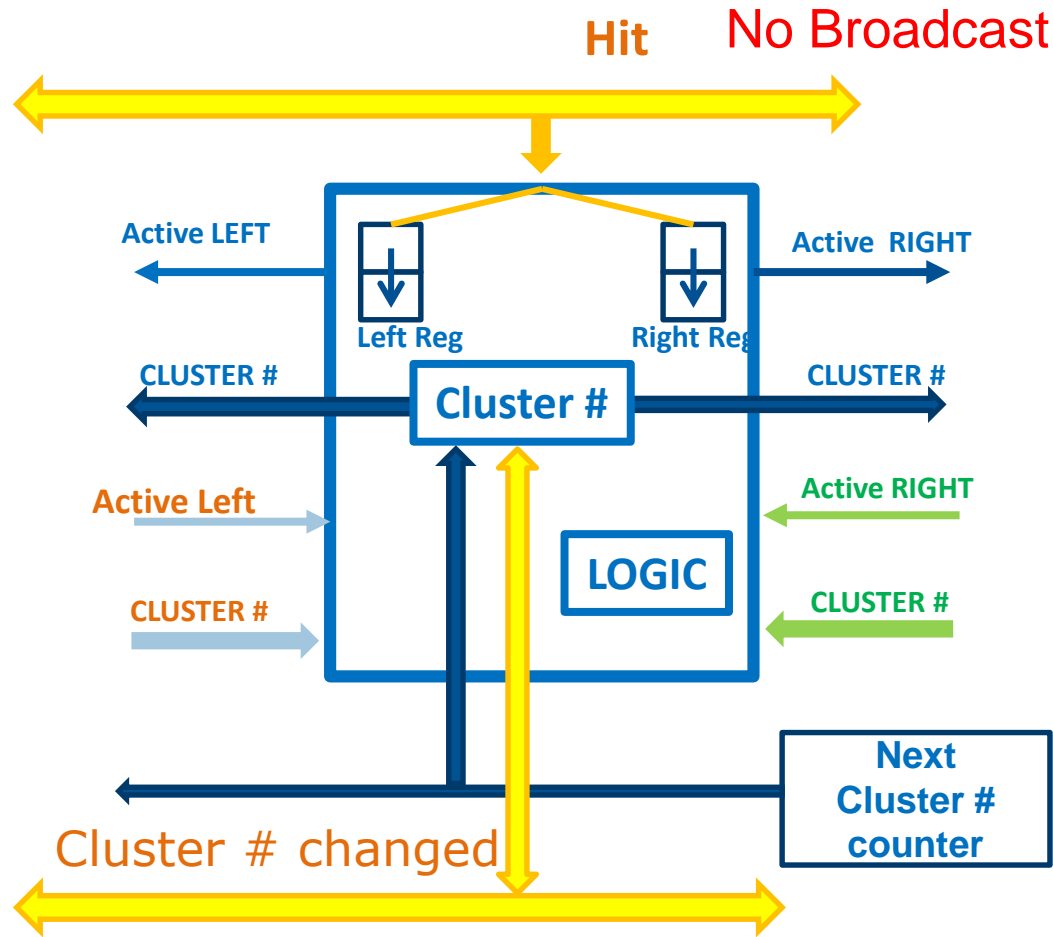


Cluster memory

Addr.	Value
1	-
2	-
3	-
4	-
5	-

When FSM changes Cluster# from B to A it broadcast change to all FSMs. Active FSMs with cluster number B also change its' cluster number to A.

# Broadcast Cluster Number change (2)



```

. 1 . . . . . . . . . . . . . . . . .
. . 1 . . . . . . . . . . 2 . . . . .
. . . 1 . . 3 . . . . . . 2 . . . . .
. . . . 1 . 3 3 . . . 2 . . . . .
. . . . . 1 . 3 . 2 . . . . . . . . .
. . . . . . . X X . . . . . . . . .
. . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . .

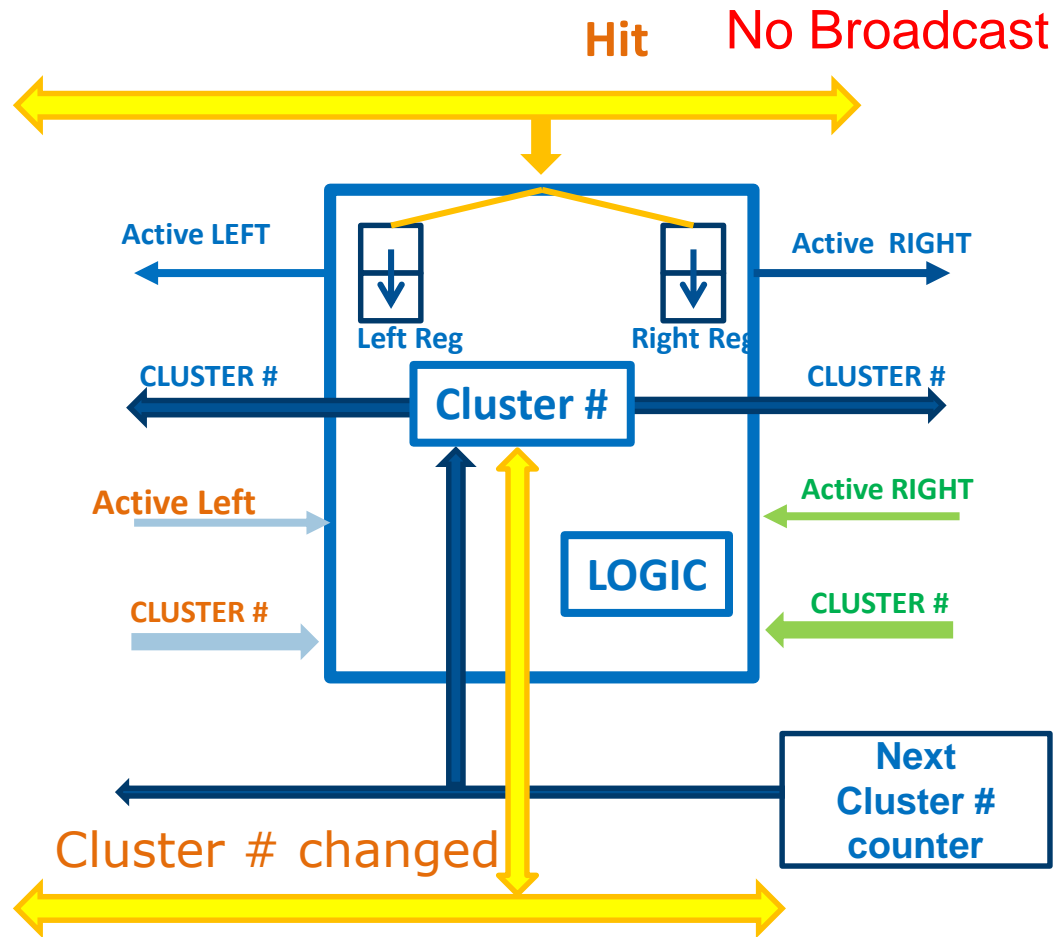
```

Cluster memory

Addr.	Value
1	1
2	2
3	1
4	-
5	-

When FSM changes Cluster# from B to A it broadcast change to all FSMs. Active FSMs with cluster number B also change its' cluster number to A.

# Broadcast Cluster Number change (3)



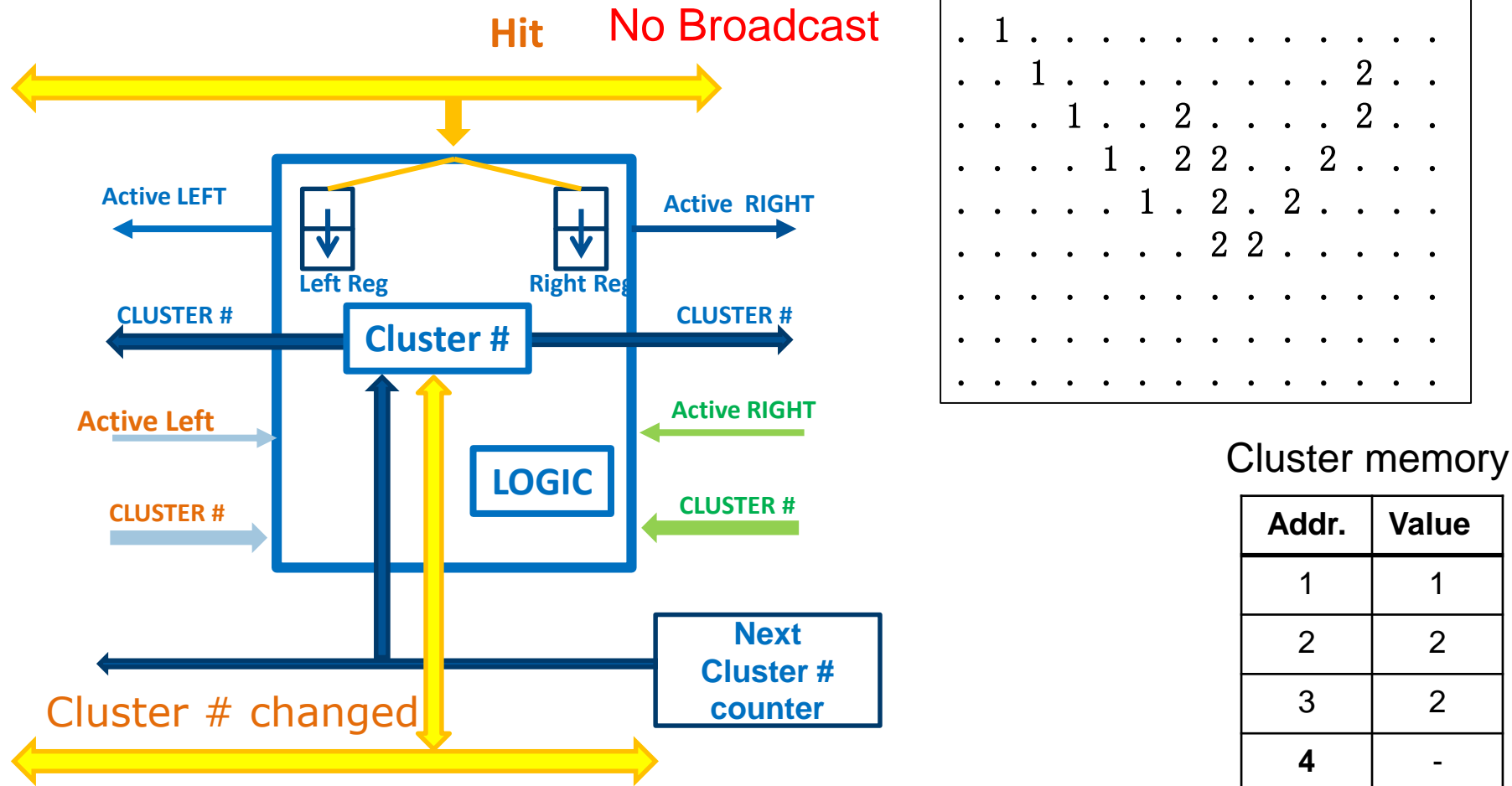
.	1	.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	1	.	.	.	.	.	.	.	2	.	.	.	.
.	.	.	1	.	.	3	.	.	.	.	2	.	.	.
.	.	.	.	1	.	3	3	.	.	2	.	.	.	.
.	.	.	.	.	1	.	3	.	2	.	.	.	.	.
.	.	.	.	.	.	.	2	2	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.

Cluster memory

Addr.	Value
1	1
2	2
3	2
4	-
5	-

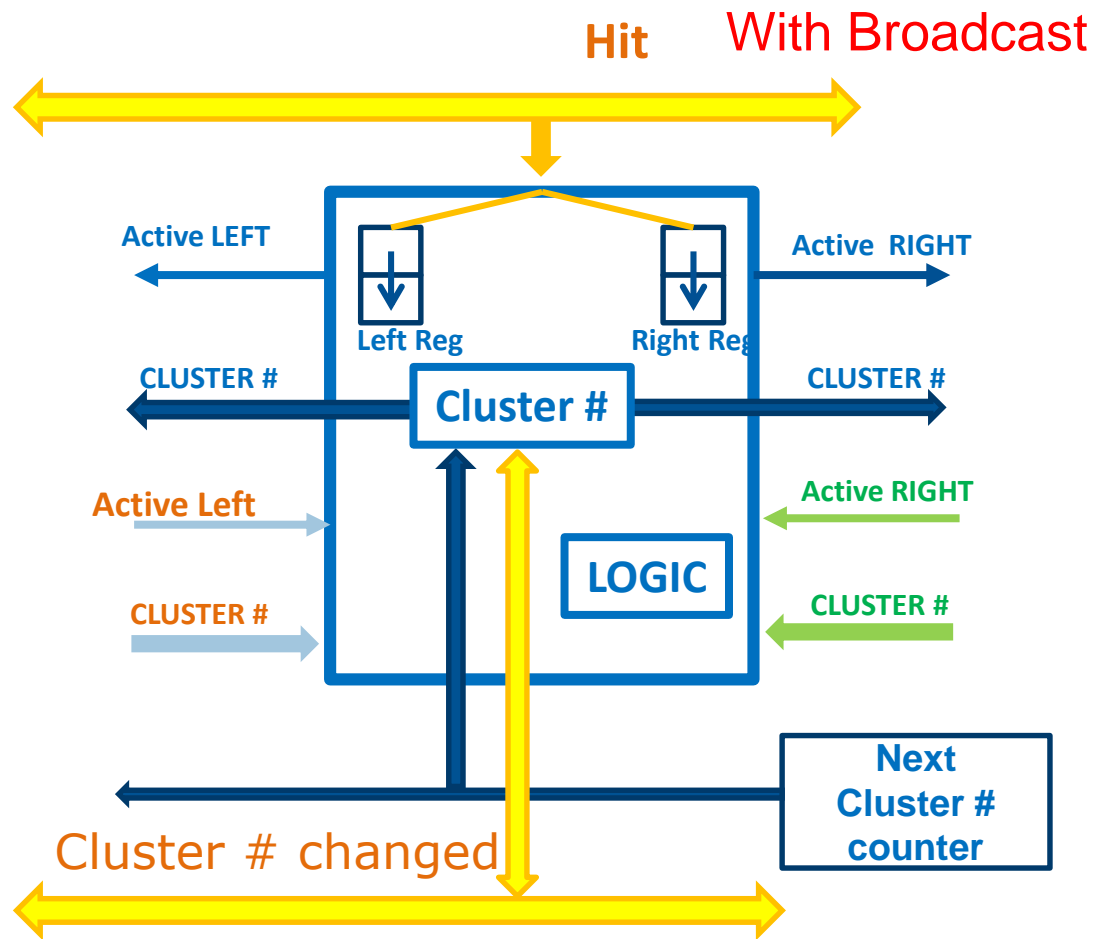
When FSM changes Cluster# from B to A it broadcast change to all FSMs. Active FSMs with cluster number B also change its' cluster number to A.

# Broadcast Cluster Number change (4)



When FSM changes Cluster# from B to A it broadcast change to all FSMs. Active FSMs with cluster number B also change its' cluster number to A.

# Broadcast Cluster Number change (5)



.	1	.	.	.	.	.	.	.	.	.	.	.
.	.	1	.	.	.	.	.	.	.	2	.	.
.	.	.	1	.	.	3	.	.	.	.	2	.
.	.	.	.	1	.	3	3	.	.	2	.	.
.	.	.	.	.	1	.	1	.	2	.	.	.
.	.	.	.	.	.	.	X	X	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.

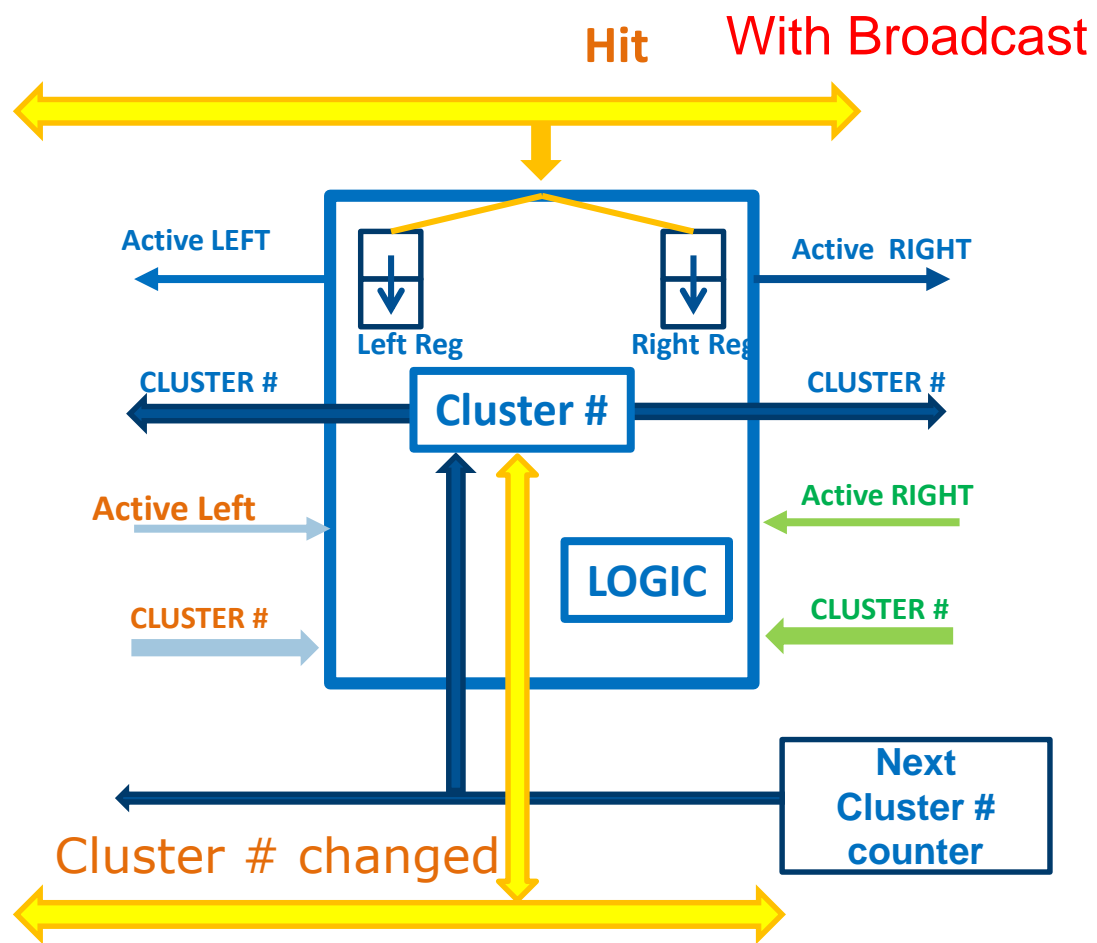
Cluster memory

Addr.	Value
1	1
2	2
3	1
4	-
5	-

When FSM changes Cluster# from B to A it broadcast change to all FSMs. Active FSMs with cluster number B also change its' cluster number to A.



# Broadcast Cluster Number change (6)



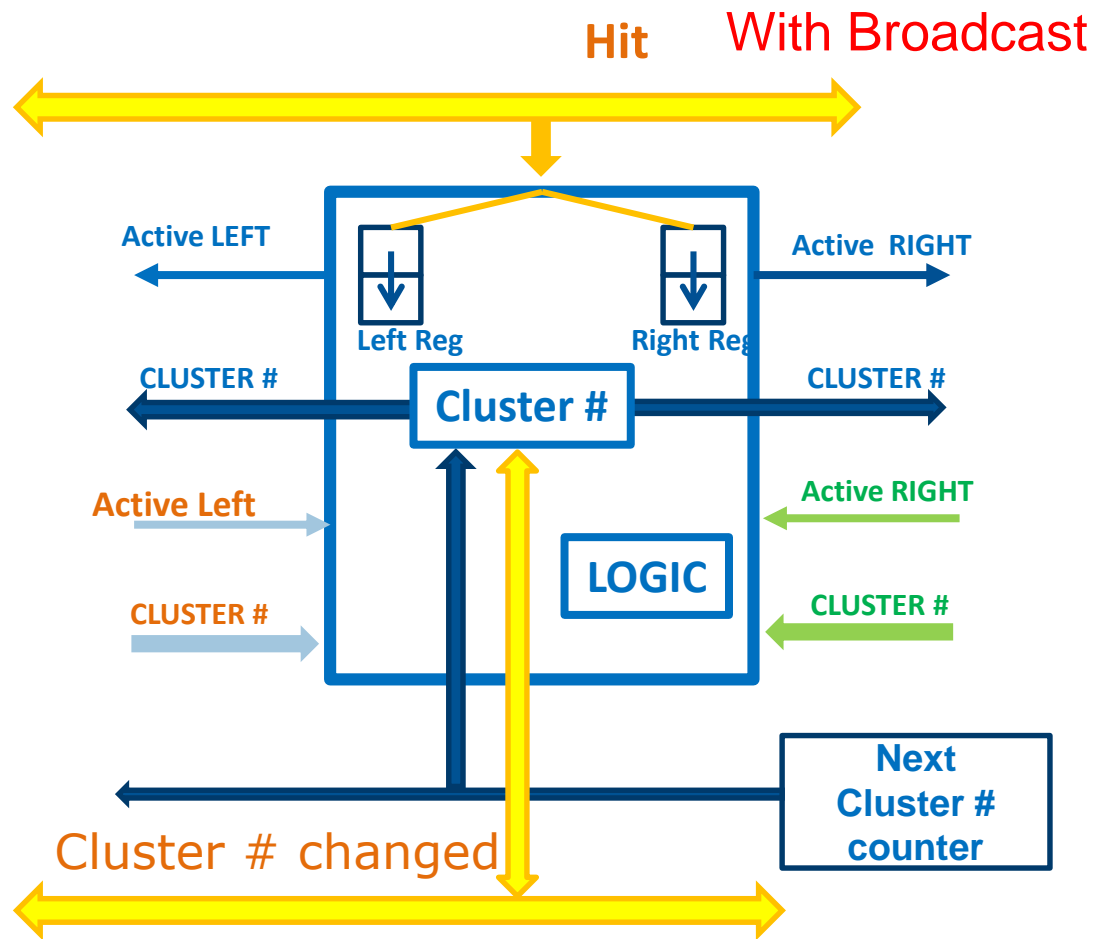
.	1	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	1	.	.	.	.	.	.	.	2	.	.	.	.	.	.	.	.	.	.
.	.	.	1	.	.	3	.	.	.	.	2	.	.	.	.	.	.	.	.	.
.	.	.	.	1	.	3	3	.	.	2	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	1	.	1	.	2	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	1	1	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.

Cluster memory

Addr.	Value
1	1
2	1
3	1
4	-
5	-

When FSM changes Cluster# from B to A it broadcast change to all FSMs. Active FSMs with cluster number B also change its' cluster number to A.

# Broadcast Cluster Number change (6)



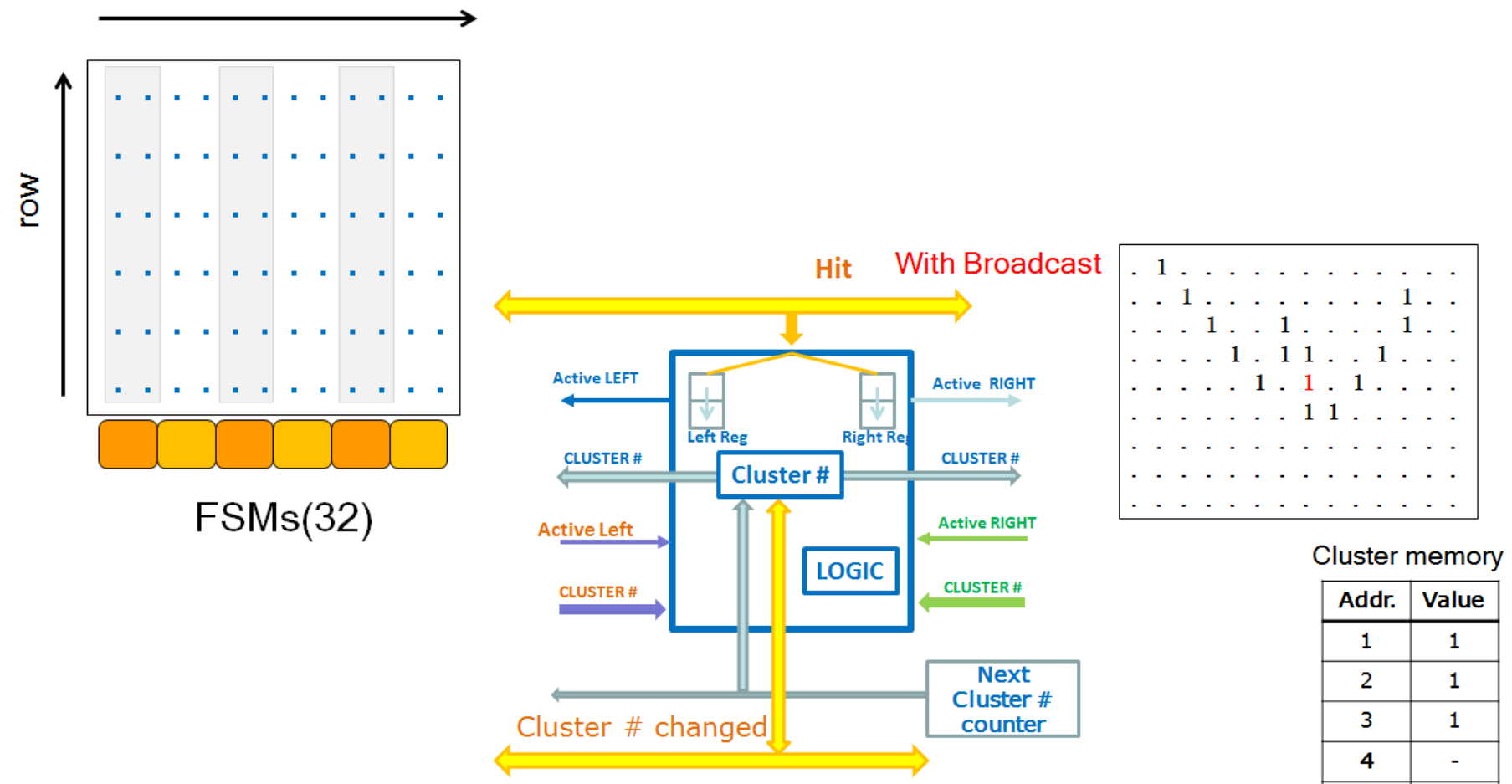
.	1	.	.	.	.	.	.	.	.	.	.	.
.	.	1	.	.	.	.	.	.	1	.	.	.
.	.	.	1	.	.	1	.	.	.	1	.	.
.	.	.	.	1	.	1	1	.	.	1	.	.
.	.	.	.	.	1	.	1	.	1	.	.	.
.	.	.	.	.	.	.	1	1	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.

Cluster memory

Addr.	Value
1	1
2	1
3	1
4	-
5	-

When FSM changes Cluster# from B to A it broadcast change to all FSMs. Active FSMs with cluster number B also change its' cluster number to A.

# TUM Clustering algorithm



When FSM changes Cluster# from B to A it broadcast change to all FSMs. Active FSMs with cluster number B also change its' cluster number to A.

# Storing hits after clustering

Cluster addresses table

Cluster #	Address of last hit	Address of first hit
1	0	0
2	1	1
...	...	...

Hit memory after clustering

Addr.	Cluster #	Address of previous hit
0	1	-
1	2	-
2		
3		
4		
5		
6		
7		
8		

## Processing one hit :

1. Evaluating Hit cluster number and writing to Hit Memory
2. Hit information includes cluster # and address of previous hit of the cluster.
3. Update table with addresses of the last hits
4. Increment address counter

# TUM Storing hits after clustering

Cluster addresses table

Cluster #	Address of last hit	Address of first hit
1	0	0
2	1	1
...	...	...

Hit memory after clustering

Addr.	Cluster #	Address of previous hit
0	1	-
1	2	-
2	1	0
3		
4		
5		
6		
7		
8		

## Processing one hit :

1. Evaluating Hit cluster number and writing to Hit Memory
2. Hit information includes cluster # and address of previous hit of the cluster.
3. Update table with addresses of the last hits
4. Increment address counter

Cluster addresses table

Cluster #	Address of last hit	Address of first hit
1	2	0
2	1	1
...	...	...

Hit memory after clustering

Addr.	Cluster #	Address of previous hit
0	1	-
1	2	-
2	1	0
3		
4		
5		
6		
7		
8		

## Processing one hit :

1. Evaluating Hit cluster number and writing to Hit Memory
2. Hit information includes cluster # and address of previous hit of the cluster.
3. Update table with addresses of the last hits
4. Increment address counter

# Storing hits after clustering

Cluster addresses table

Cluster #	Address of last hit	Address of first hit
1	8	0
2	7	1
...	...	...

Hit memory after clustering

Addr.	Cluster #	Address of previous hit
0	1	-
1	2	-
2	1	0
3	1	2
4	2	1
5	2	4
6	1	3
7	2	5
8	1	6

## Processing one hit :

1. Evaluating Hit cluster number and writing to Hit Memory
2. Hit information includes cluster # and address of previous hit of the cluster.
3. Update table with addresses of the last hits
4. Increment address counter

Cluster addresses table

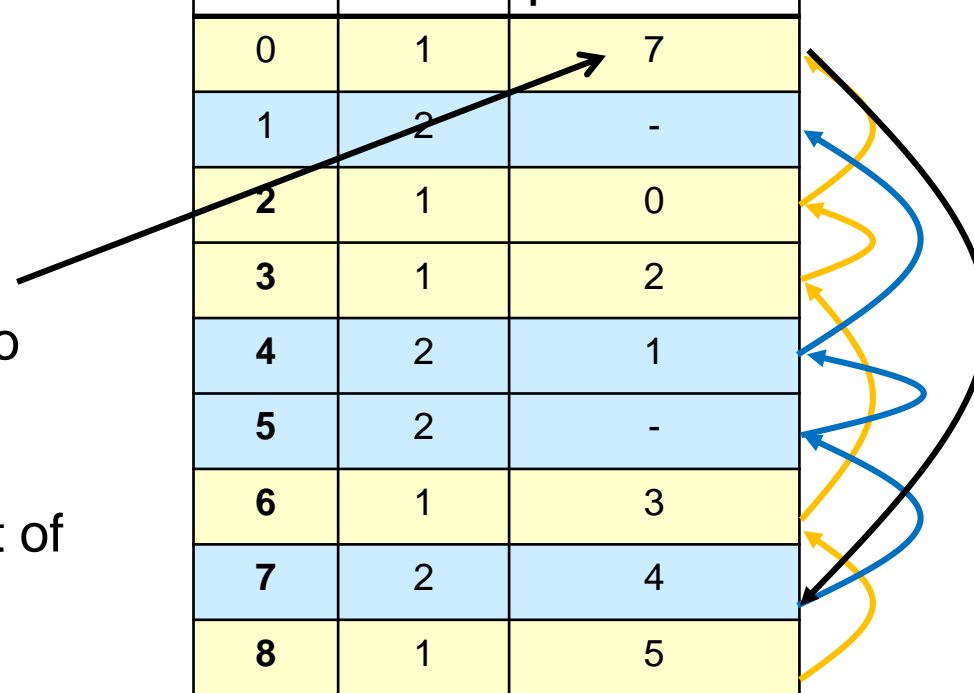
Cluster #	Address of last hit	Address of first hit
1	8	0
2	7	
...	...	...

**Merging clusters 1 and 2 :**  
 write address of last hit of cluster 2 to  
 address field of first hit of cluster 1

declare first hit of cluster 2 as first hit of  
 cluster 1

Hit memory after clustering

Addr.	Cluster #	Address of previous hit
0	1	7
1	2	-
2	1	0
3	1	2
4	2	1
5	2	-
6	1	3
7	2	4
8	1	5





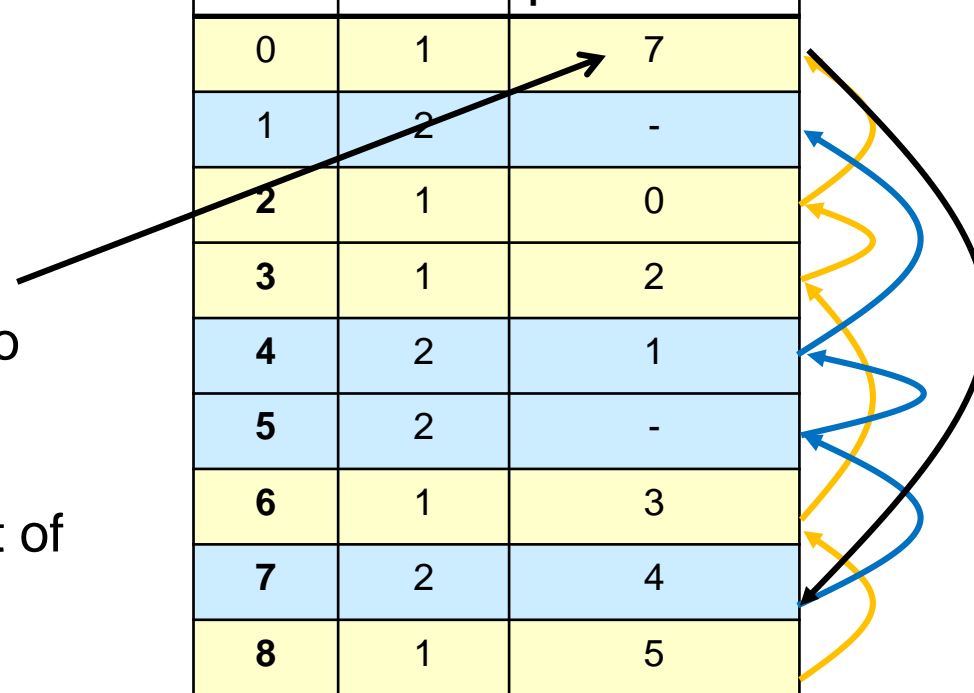
Cluster addresses table

Cluster #	Address of last hit	Address of first hit
1	8	1
2	7	
...	...	...

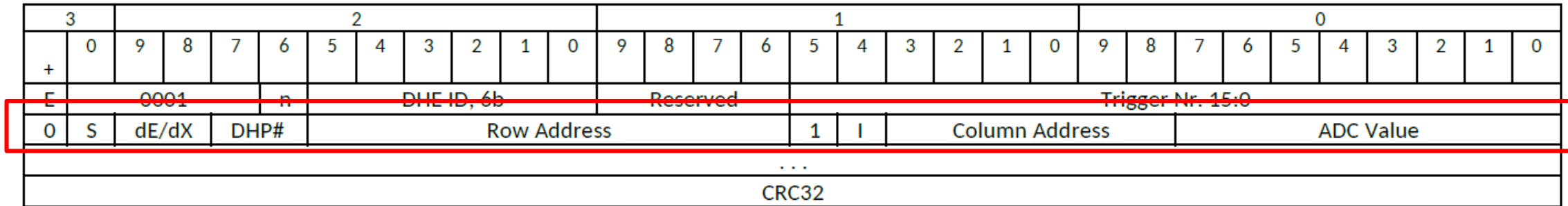
Hit memory after clustering

Addr.	Cluster #	Address of previous hit
0	1	7
1	2	-
2	1	0
3	1	2
4	2	1
5	2	-
6	1	3
7	2	4
8	1	5

**Merging clusters 1 and 2 :**  
 write address of last hit of cluster 2 to  
 address field of first hit of cluster 1  
  
 declare first hit of cluster 2 as first hit of  
 cluster 1



## DHP Clustering Engine Frame



- S = 1 – start of cluster
- dE/dx – 2 bit information to encode NN cluster evaluation as slow pion
- DHP #(2b), Row#(10b) – starting row of cluster

First hit

- I = 0, Column# (6bits), Row # unchanged
- I = 1, Column# (6bits), Row # decremented

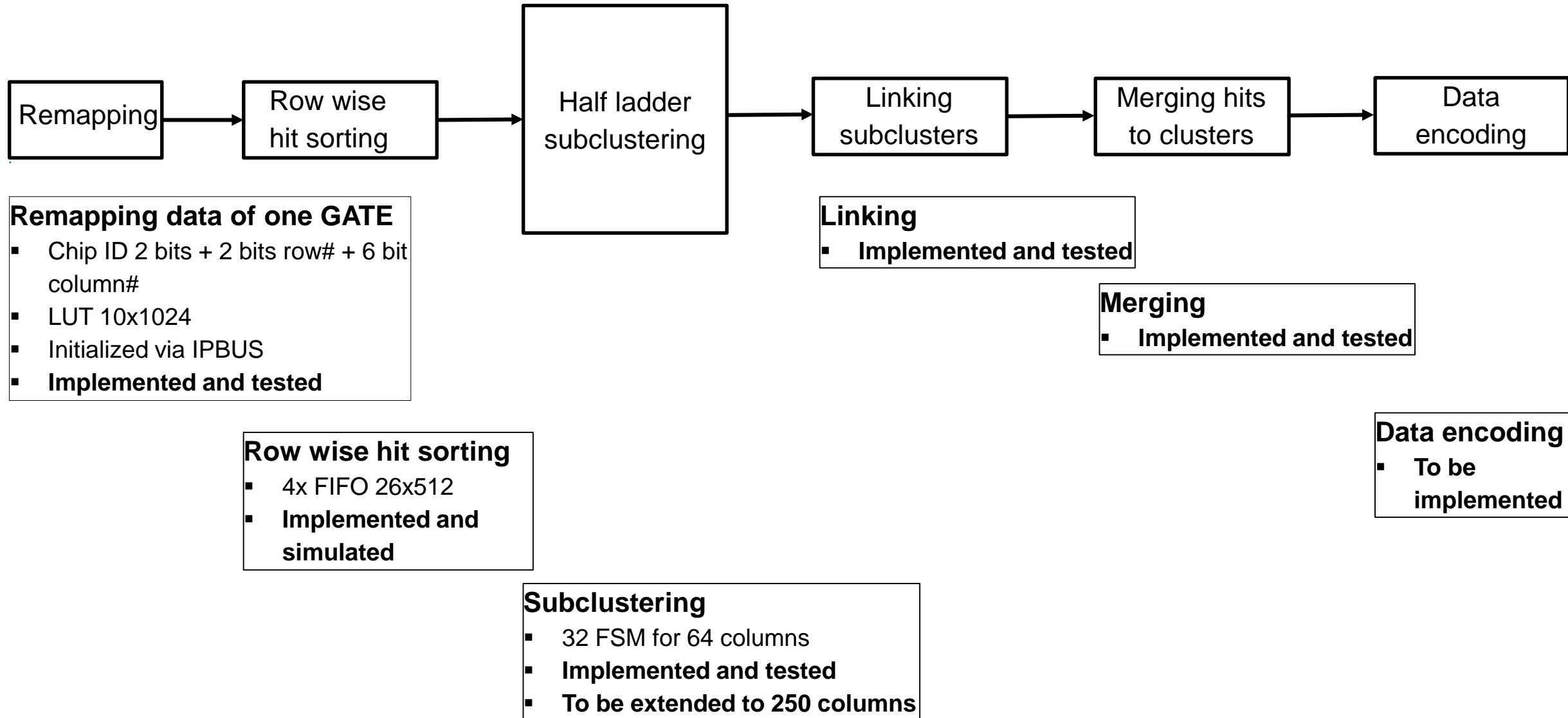
Following hits

- I = 0, Column#(1 + 5 bits) – change of column number relatively to previous hit, Row # unchanged
- I = 1, Column#(1 + 5 bits) – change of column number relatively to previous hit, Row # decremented

If change of column number > ± 31

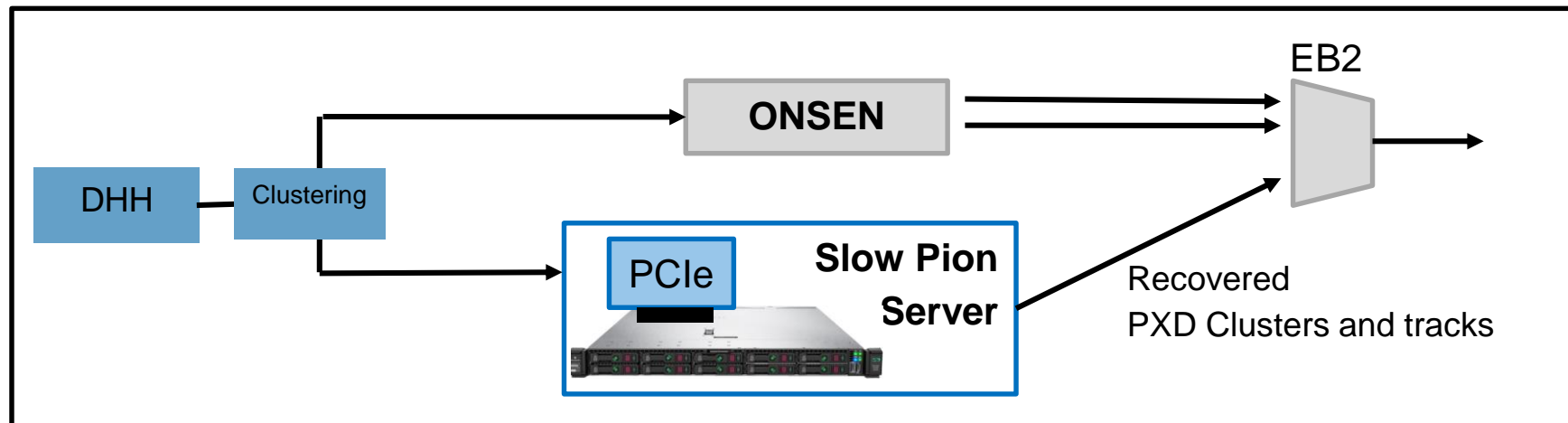
- S = 0 – start of cluster
- dE/dx – 2 bits not used
- DHP #(2b), Row#(10b) – row #
- ...

# Clustering Algorithm Status



# Data Flow I

## 1. Standalone pion track reconstruction

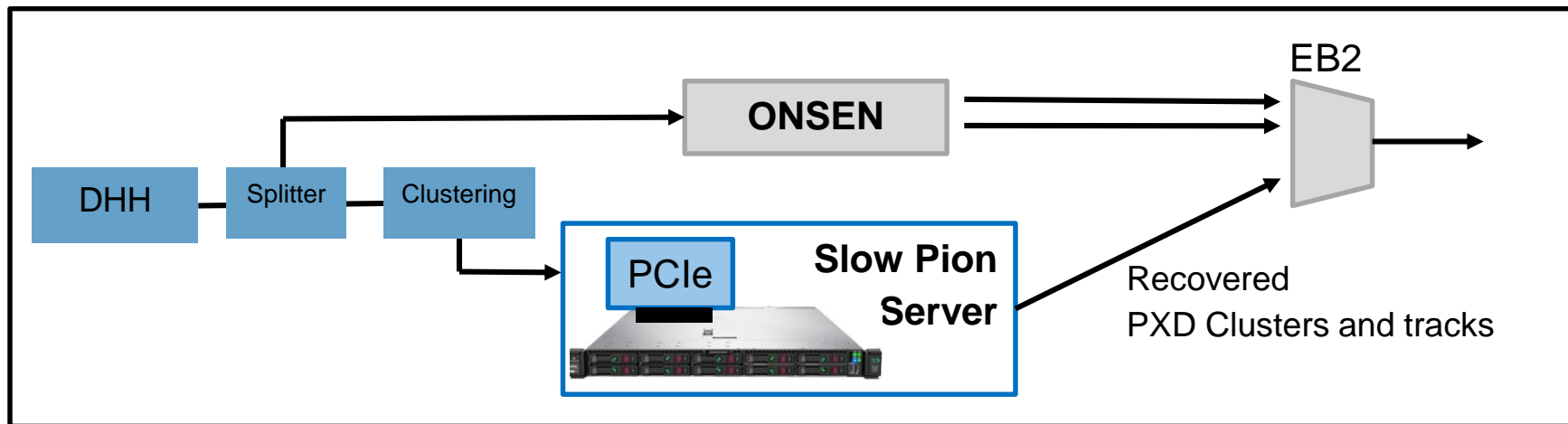


ONSEN – no change of data format

- Concern from Giessen group about change of ONSEN firmware to support clustering format
- Rigid firmware architecture and DAQ interface
  - Necessity to include X-switch to reorder detector information
  - Necessity to get event from DHH before HLT ROIs
  - Not possible to change number of DHH-ONSEN links without recompilation HLT ROI distribution
  - ...

# Data Flow I

## 1. Standalone pion track reconstruction



No clustering data to ONSEN – no change of data format, very small delay of data

- Concern from Giessen group about change of ONSEN firmware to support clustering format
- Rigid firmware architecture and DAQ interface
  - Necessity to include X-switch to reorder detector information
  - Requirement to get event from DHH before ROIs from HLT
  - Not possible to change number of DHH-ONSEN links without recompilation HLT ROI distribution
  - Big development to integrate ONSEN to DAQ
  - ...

# DHH Data Splitter

## DHC

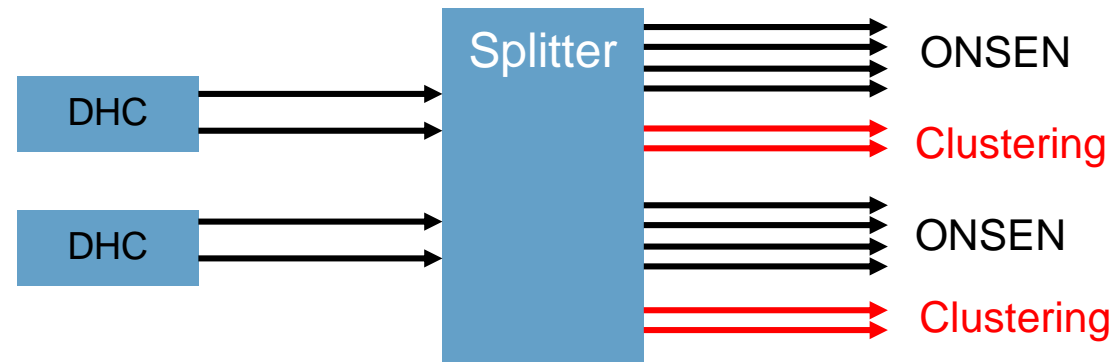
- 2x2 links @6.5 Gbps

## ONSEN

- 2x4 links @6.5 Gbps

## Clustering FPGA

- 4 links @6.5 Gbps



# FPGA Card for Clustering

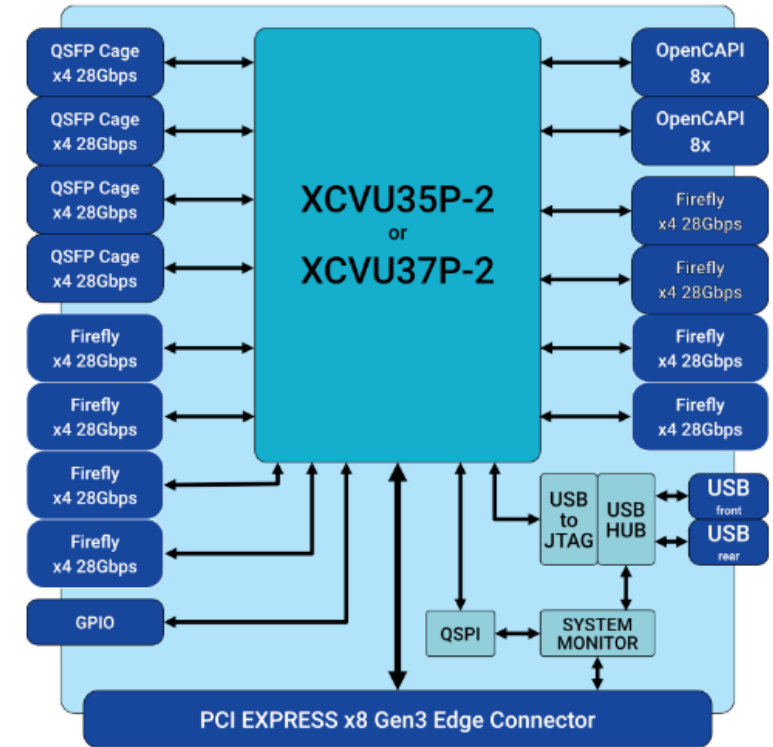
## PCIE Card from Alpha-Data

- XCV35P-2/XCV37P-2
- 48/64 high speed links 16-28 Gbps
- 2x 4GB HBM Gen2 memory (32 AXI Ports provide 460GB/s Access Bandwidth)
- 6x PCI Express x16 Gen3 / x8 Gen 4 cores (COX Capable) <https://tum-conf.zoom.us/j/69020128146>
- To be used as additional serial links

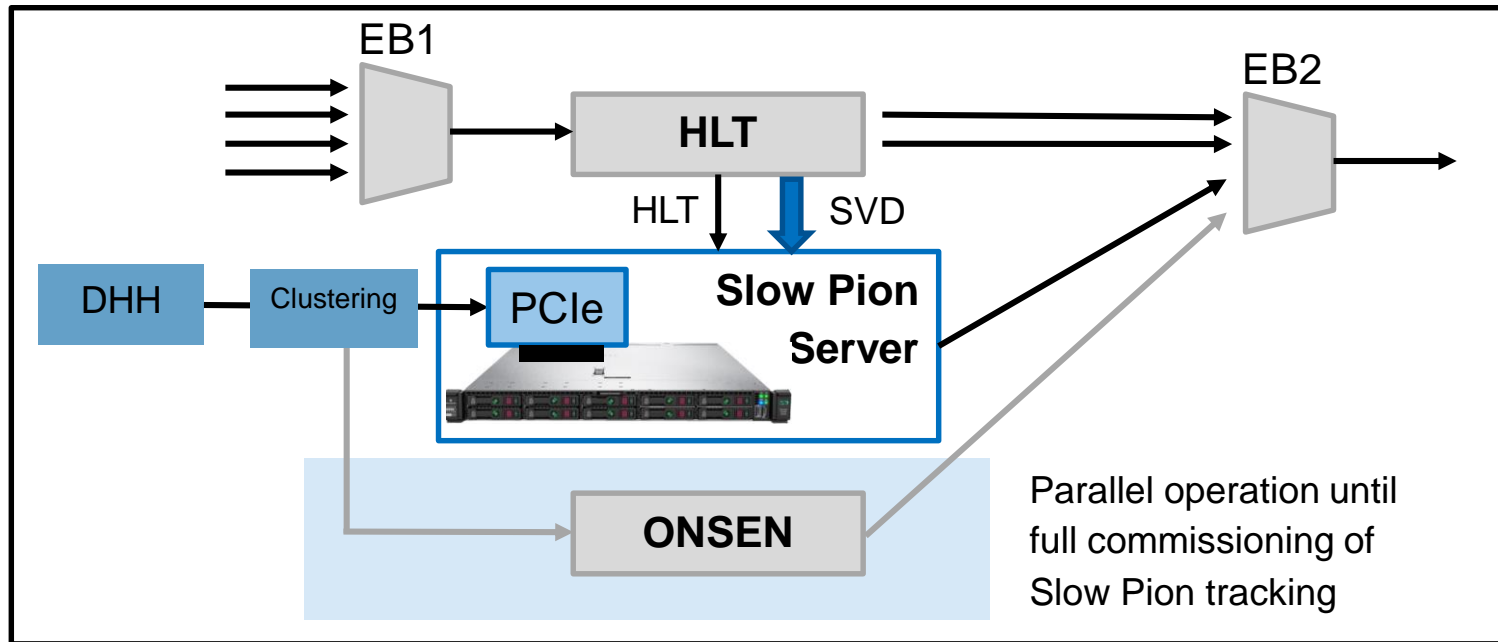
## XCV37P vs Virtex6-130T

- |              |                       |                    |
|--------------|-----------------------|--------------------|
| ▪ LUT        | 1300k/75k = ~20       | } ~36 times bigger |
| ▪ FF         | 2600/160k = ~16       |                    |
| ▪ Speed gain | 400MHz vs 200MHz => 2 |                    |

- One card will be sufficient for full PXD clustering and event building
- Will be available for tests in December



# Data Flow II



## Slow Pion Server

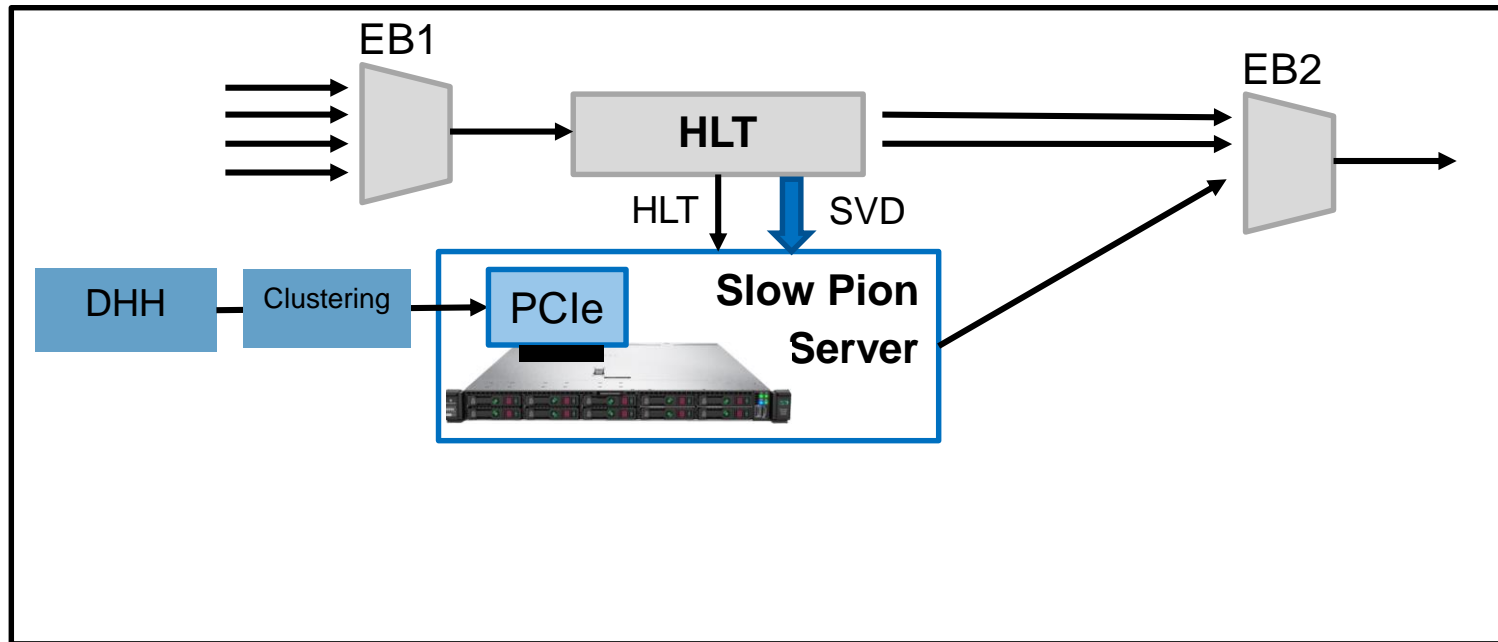
- 128-256 GB RAM
- 10/25 GEthernet
- BASF2 tracking optimized for fast reconstruction

## Advantage :

- Overcome ONSEN memory limit
- Standard data flow
- Minimum hardware



# Data Flow III



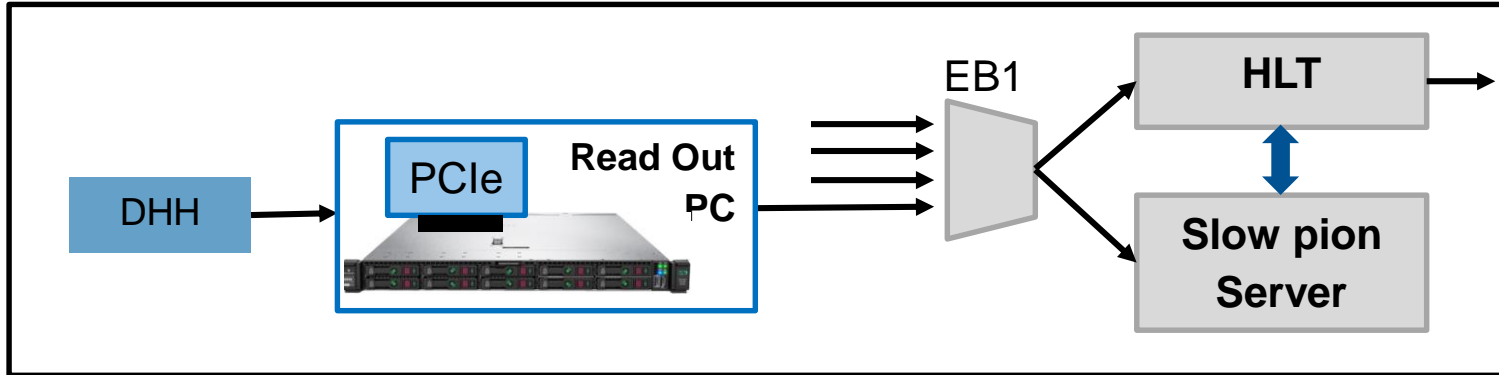
## Slow Pion Server

- 256 GB RAM to buffer PXD data
- 10/25 G Ethernet
- BASF2 tracking optimized for fast reconstruction

## Advantage :

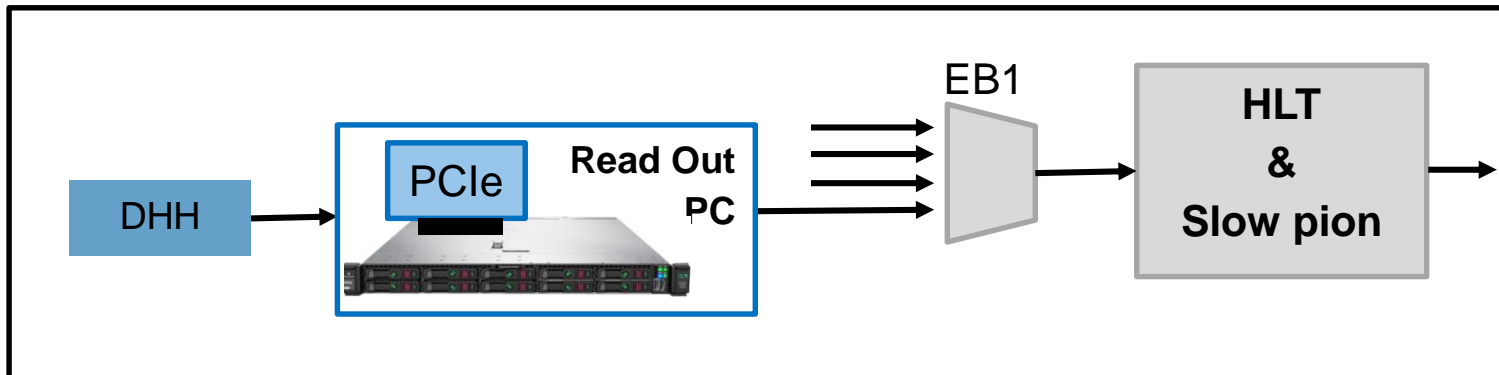
- Overcome ONSEN memory size limit
- Standard data flow

# Data Flow IV



Advantage :

- Overcome limited memory in ONSSEN
- Standard data flow and data processing infrastructure
- **No EB2**
- If PXD collaboration and DAQ group approves it can be considered for PXD2 installation **wo slow pion track reconstruction**



Disadvantage :

- Higher data rate
- PXD latency to be compensated

Advantage for both : overcome limited ONSSEN memory size and allow to extend HLT processing time

# Summary conclusions

- FPGA clustering algorithm advanced and shall be completed before PXD2
- Clustering to be tested and commissioning
- Main approach to integrate clustering with slow particle tracking