DHH status report

Igor Konorov
TUM Physics Department E18

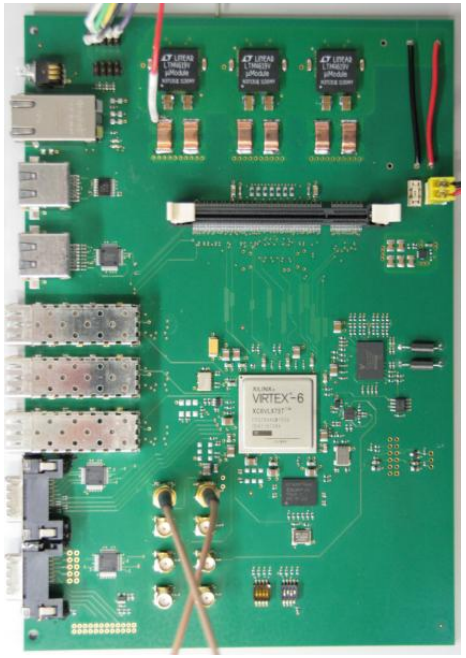
13th International Workshop on DEPFET Detectors and Applications

Ringberg, 12-15 June 2013

Overview

- EMCM tests
- DHH status
  - DHH AMC module
  - DHH Carrier Card
- Preparation for DESY test beam
- Plans for production
- Clustering algorithm
  - Introduction
  - Architecture
  - FPGA resources

DHH prototype modules



DHH prototype

- DHP Hybrid, FTSW and ONSEN modules
- integrated into EPICS slow control
- Operated in May testbeam at DESY

- Problem with synchronization:

Data scrambled , under investigation now

- One DHH module at DESY for further slow control development
- One module at HLL in preparation for EMCM tests

EMCM test environment

System aspects:

- DHH only ReadOut
- EPICS slow control
- DHH acquires data via IPbus (Ethernet) , data rate 1 MB/s
- Synchronous with external trigger **full**/sparse frame readout !!!

DAQ & DQM options

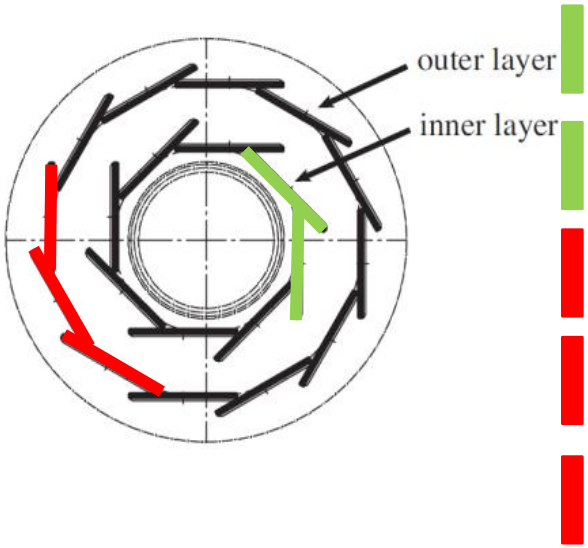
1. Bonn DAQ with existing analysis tools

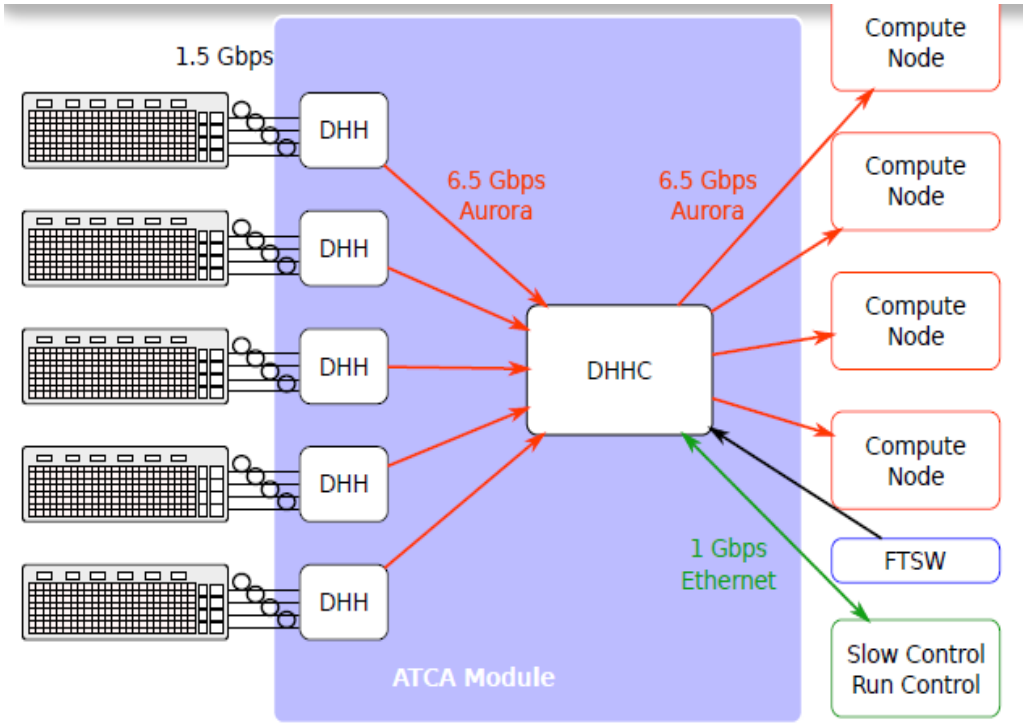
2. EPICS based Run Control plus Benjamin's test beam analysis software
3. EPICS based Run Control plus stripped version of BASF2

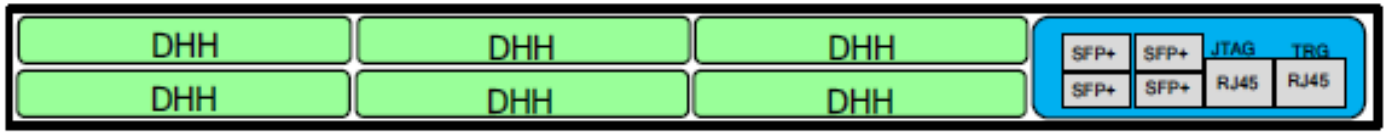
DHH system

DHH/DHHC

Readout scheme

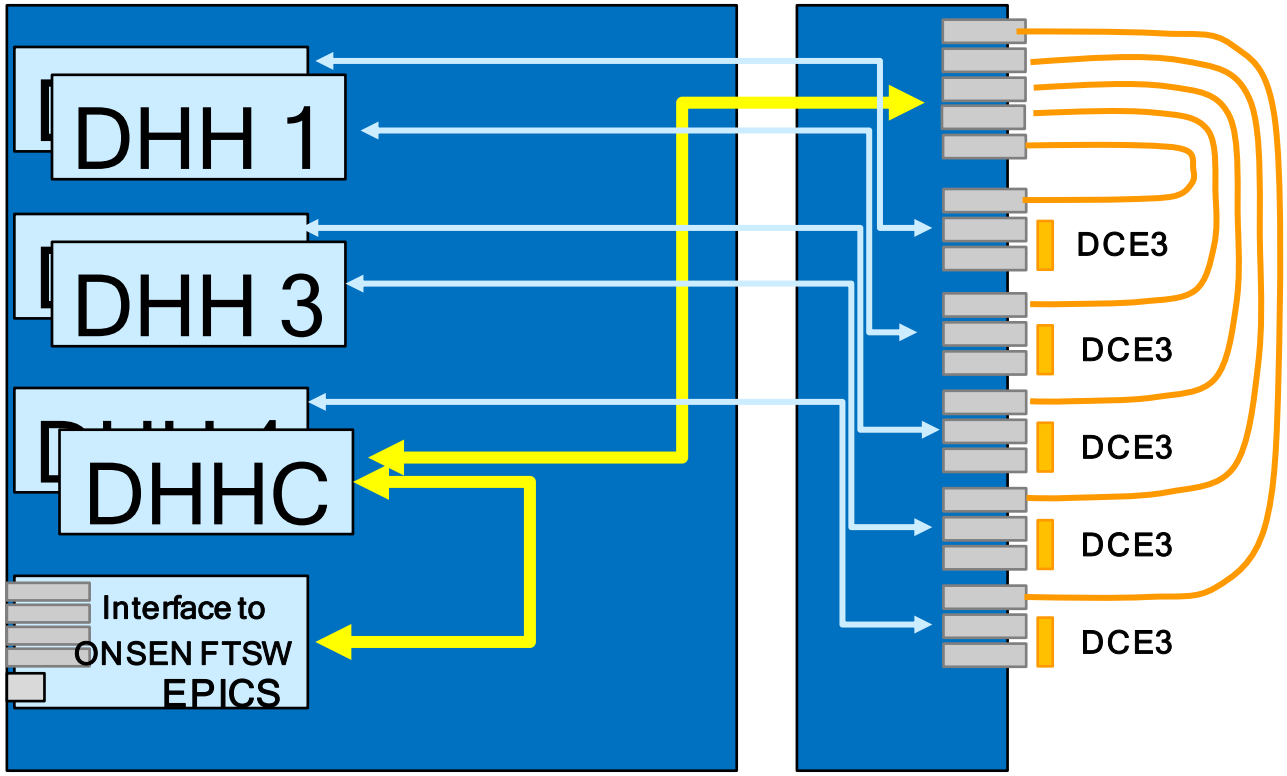






ATCA module

ATCA DHH Carrier Card



ATCA Carrier Card

RTM

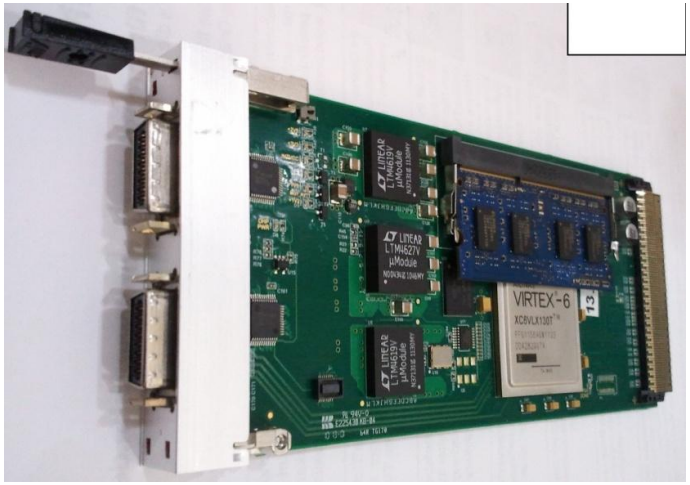
DHHC ⇔ DHHs serial link

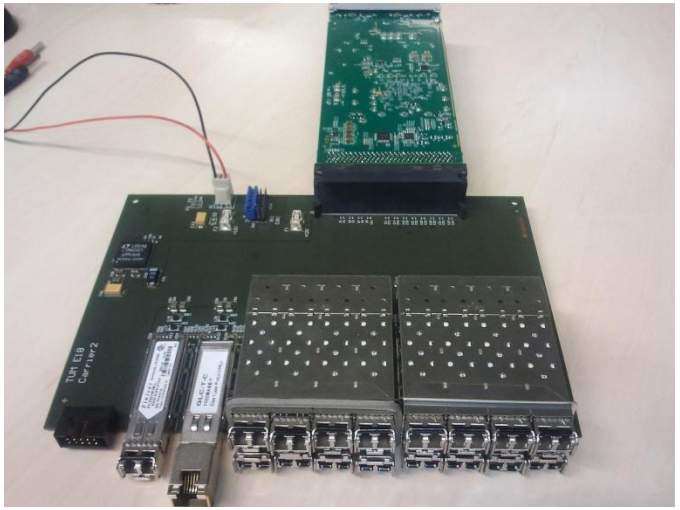
- FTSW signals
- Slow Control(IPBus)
- DATA

Two options for DCE3

- SODIMM socket on DHH or
- External location

DHH/DHHC final design





Final design

- AMC module, ATCA standard
- VIRTEX6 XC6VLX130T FPGA
- 2GB DDR3
- Current monitor piggy-back card
- MMC piggy-back card
- Module programmed as DHH or DHHC

- AMC connector
  - FTSW signals (DHHC)
  - JTAG signals
  - Ethernet (DHHC)
  - 16x6.25 Gbps
  - I2C shelf manager

Module has been tested with DHP test card
No problems detected

Current monitor does not exist

DESY Test beam preparation

Hardware for test beam in DESY :

- 2 x DHH modules (DHH AMC)
- 1 DHHC module (DHH AMC)
- ATCA carrier card or a test card as a backup solution
- SODIMM Adapter card for additional JTAG and control signals for Switcher via additional Infiniband cables

Status

- 6 final DHH-AMC cards to be submitted for production next week and expected for delivery in September
- ATCA Carrier Card is in design face , finish by end of July
- Plan to submit for production and test in September-October

Plans for DHH production

- 6 final DHH-AMC modules will be submitted next week
- If test is successful a full production will be launched

Should we wait for DESY test ?

- ATCA carrier cards will be submitted for production in January-February 2014

Firmware development

Most of functionality has been implemented

- DHH
  - Synchronization with Trigger of full frame readout
  - Handling overlapping events
  - Interface to DCE3

- DHHC
  - Interface between FTSW and DHHs
  - Data multiplexer and sub-event building
  - Interface between FTSW and DHHs

Next steps

System integration is complicated process

Install system prototype in DESY soon (July?)

FTSW + Hybrid5(?) + DHH + ONSEN;

Test evolved firmware;

EPICS;

Test system interfaces

Example:

Inject data into system to DHH via IPBus and check data flow

- DHHC Event Building
- clustering
- ROI

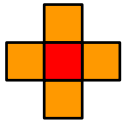
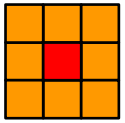
Summary

- DHH AMC card fully functioning
- Mass production scheduled for fall 2013
- DHH Carrier card is being designed
- Mass production is scheduled for February 2014
- Firmware development is advanced and will be focused on DHHC
- EMCM test and then test beam at DESY is our priority

Clustering

Clustering algorithm for FPGA

Simple requirements:

- Merging direct neighbors :  or 
- Pixel array 768x250
- Real time processing
  - 4 streams 50 10^6 pixel/second = 2 10^8 pixel/s
- Latency is not important
- Cluster data processing :
  - Center of gravity for ROI
  - Marking clusters created by low momentum particles

Literature

- **Most common case:**

Clustering for calorimeters with predefined shape 2x2,3x3, 5x5…

- **General purpose real time DCE3 clustering algorithm**
  - Parallel clustering
- **General purpose clustering(A.Annovi, M.Beretta) for ATLAS pixel detector**

Algorithm:

Each detector pixel is presented as FSM(Finite State Machine)

Detector of NxM pixels requires NxM FSMs

Clustering procedure:

Initialization , loading FSMs by hit information : EMPTY, HIT

Readout :

- external FSM selects first not empty Pixel and reads it
- SELECT signal propagates to neighboring FSMs for further readout
- this procedure is repeated till all neighboring pixels are readout

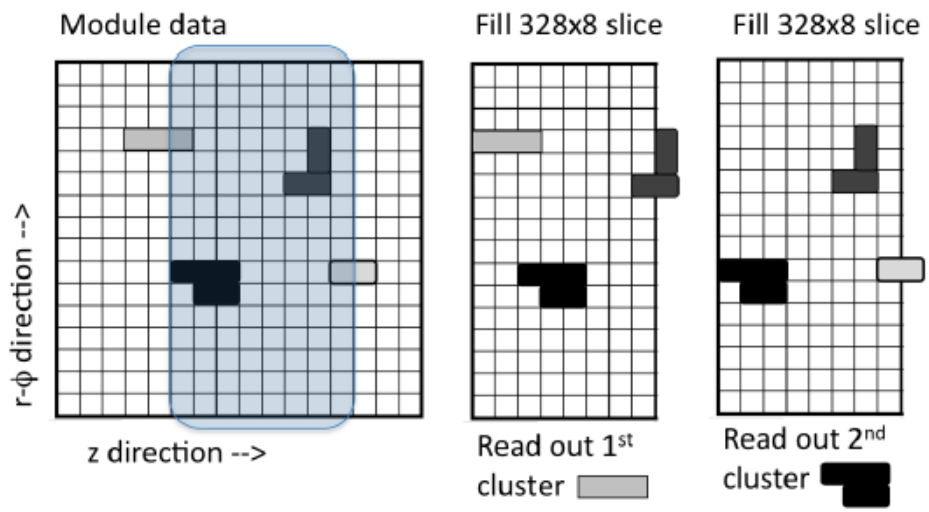
Literature

Problem: amount of hardware scales linearly with number of pixels and very fast uses up all FPGA resources:

grid size	clock period	area usage
8x8	6ns	1%
120x8	13ns	5%
32x32	13ns	6%
64x32	16ns	11%
256x8	15ns	11%
328x8	17ns	16%
120x32	20ns	21%

Table 1: Algorithm performances on a xc5vlx330 FPGA.

Solution to the problem : "Sliding Window". Window is bigger than any cluster



FPGA : XC5VLX155
Window : 328x8
30 % FPGA resources
Speed : >20 Mhits/s

New clustering algorithm for FPGA

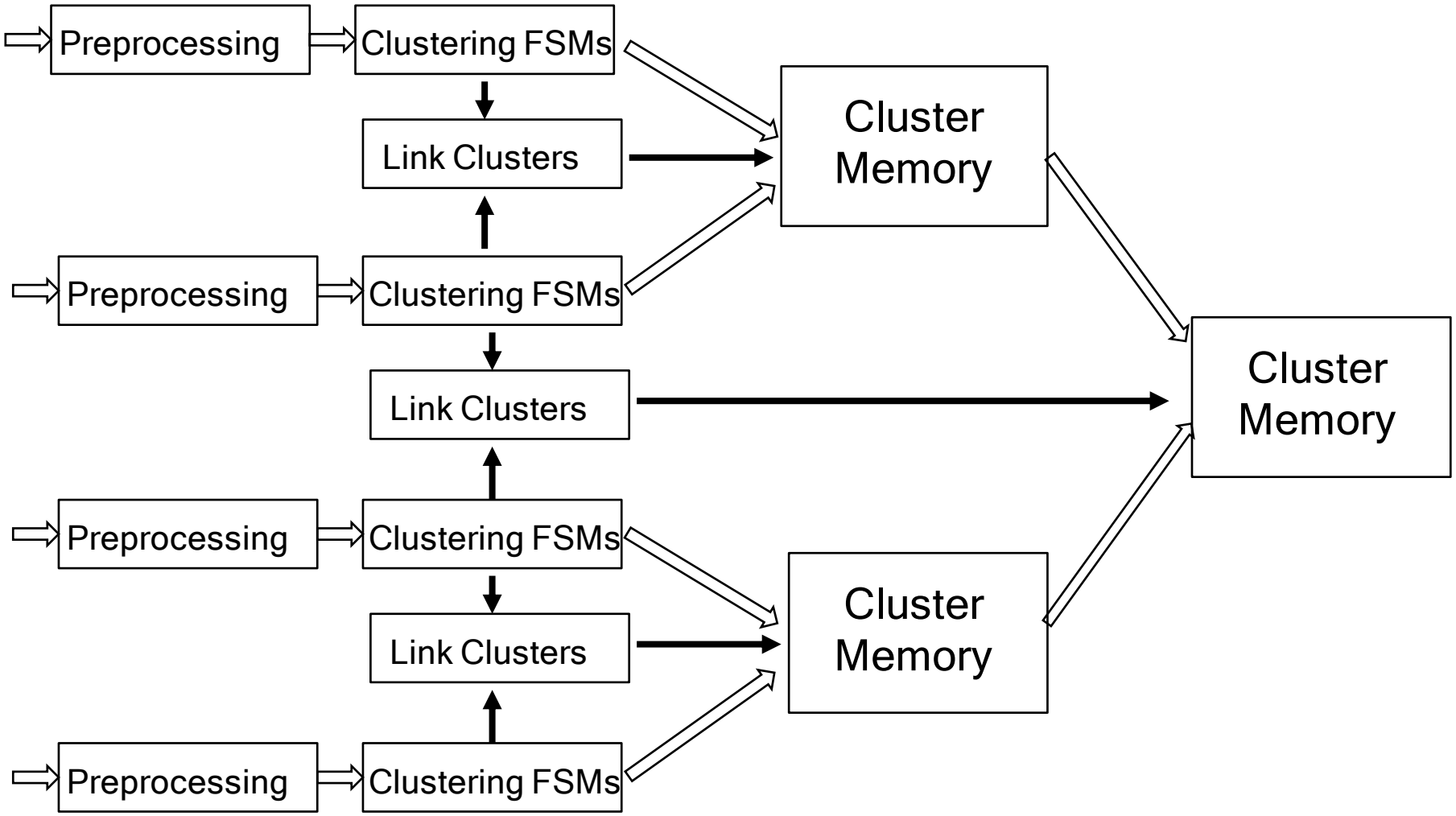
Algorithm takes advantage of DHP readout features :

- sequential data transmission
- limited data rate: less than 4 x 76 MPix/s or average 4x50 MPix/s
- 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

Algorithm data flow



Preprocessing

Remapping pixel address

Pixel row (2 LSB) and Pixel column (6 bits)

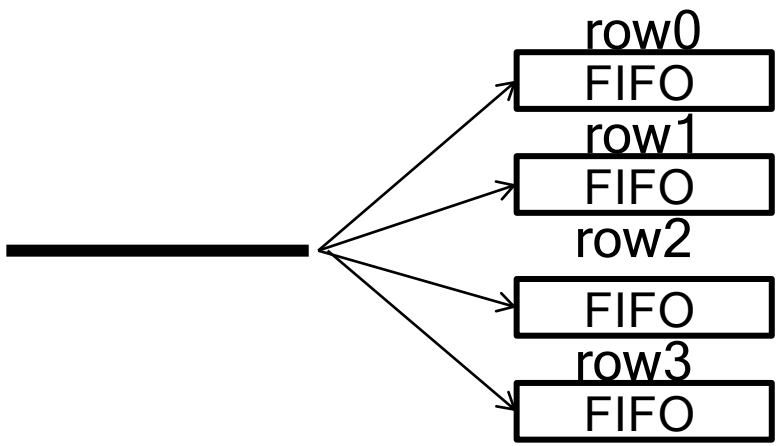
Configurable LUT 8x256

Sorting pixels to get row-wise order

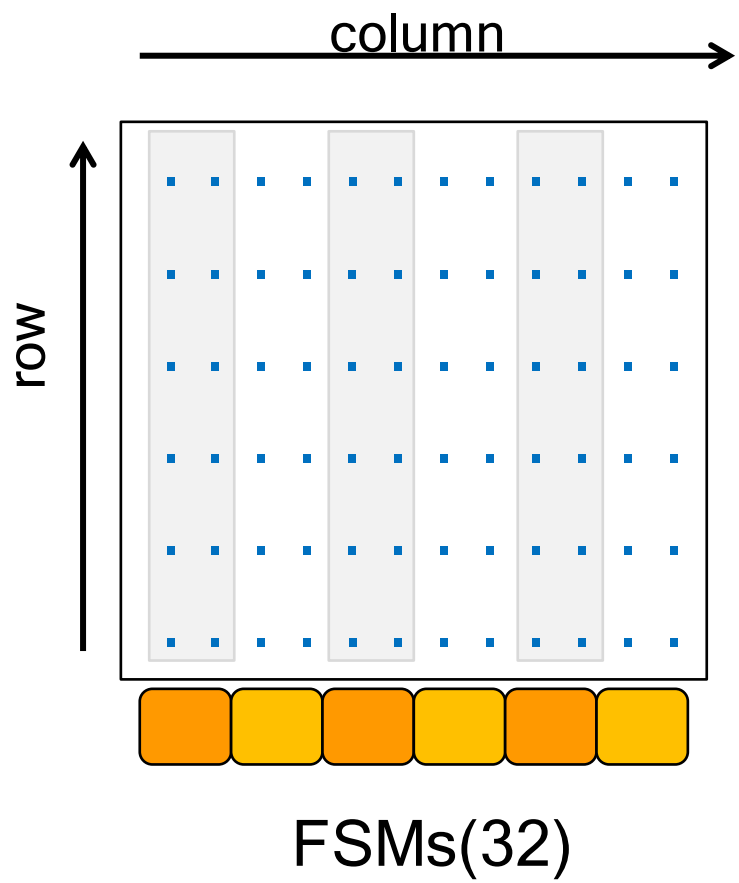
DHP zero suppression logic may mix hits within 4 rows

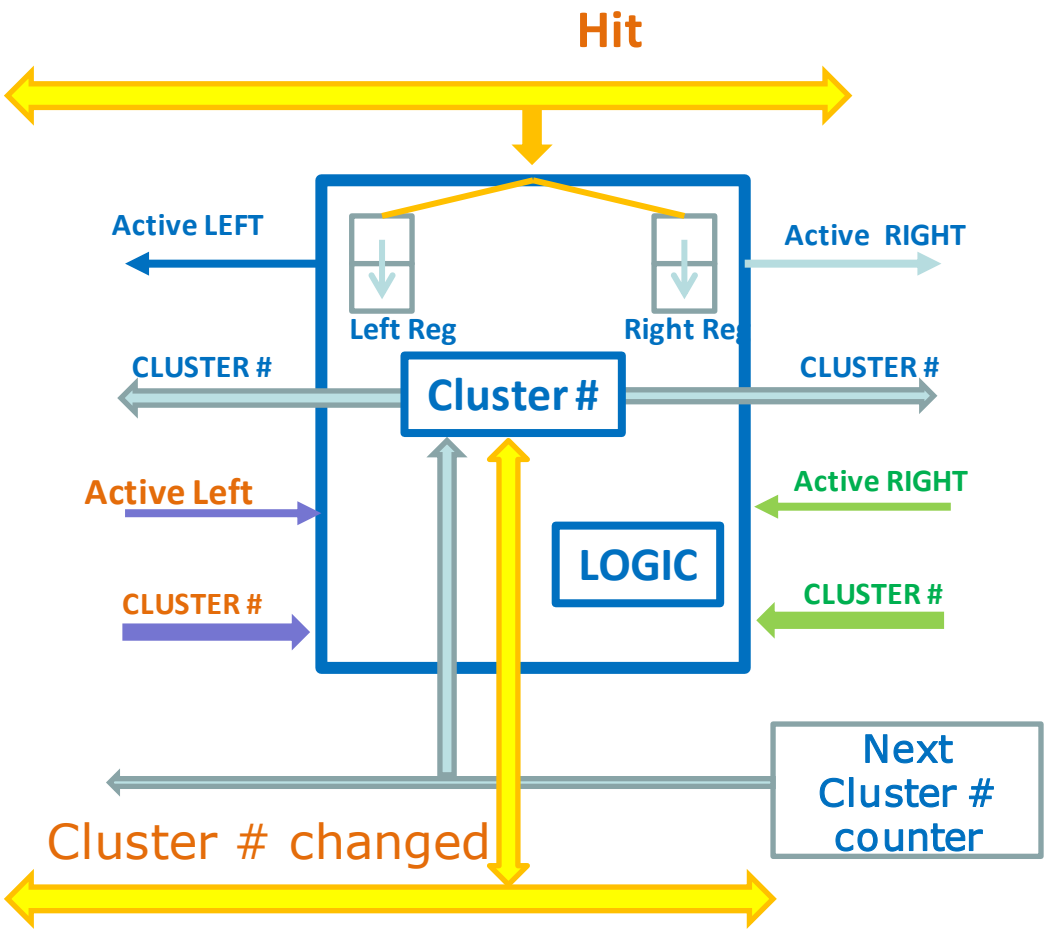
Because of processing 4x64 pixels in parallel

Sorting requires 4 FIFO 512x16



Clustering FSMs





FSM behavior

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 is described for all cases

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

Examples of FSMs actions 1

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

Next cluster counter

1

Cluster memory

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

Examples of FSMs actions 1

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

Next cluster counter

2

Cluster memory

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

Examples of FSMs actions 1

```
. . . . . . .
. . . 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
4	-
5	-

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
4	-
5	-

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	-

Examples of FSMs actions 1

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

Next cluster counter

6

Cluster memory

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

Examples of FSMs actions

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

Next cluster counter

6

Cluster memory

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

Extreme case

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

Next cluster counter

1

Cluster memory

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

Extreme case

```
. 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 . . . . . . . . . . . . . . . . . . . .
```

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

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

Cluster memory update

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

Cluster memory

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

Cluster memory

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

Cluster memory

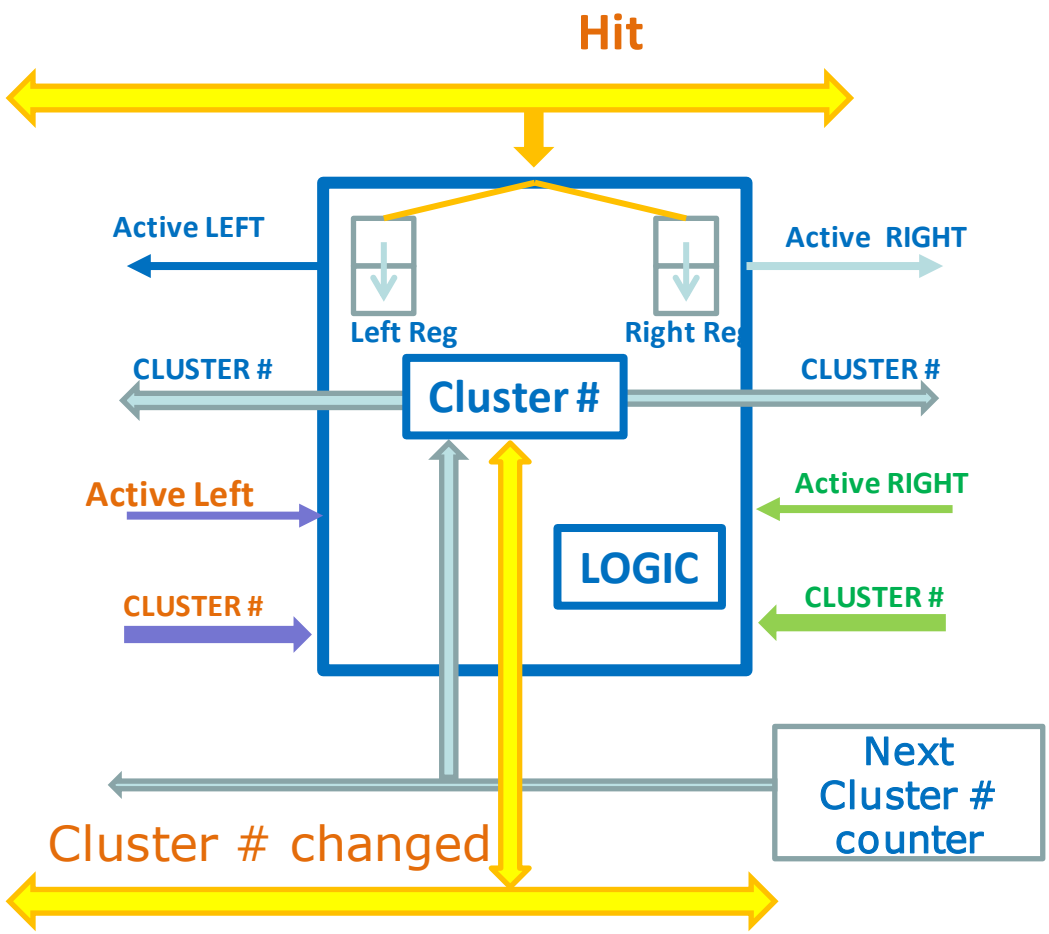
Addr.	Value
1	1
2	1
3	1
4	1
5	4
6	5
7	6
8	7
9	8

Cluster memory

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

One update takes 3 clock cycles
Using DP memory allows to reach 2 clock cycles per update

Broadcast Cluster Number change (1)



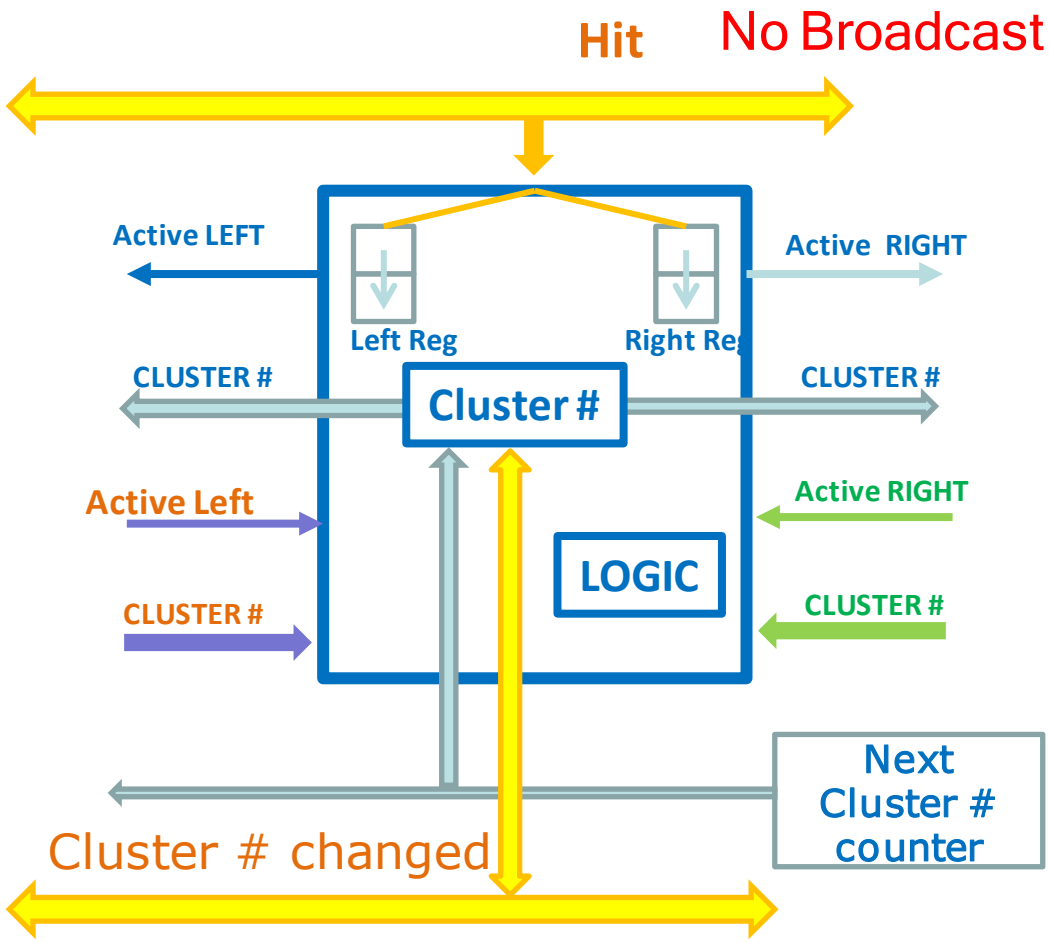

```
. X . . . . . . . . . . . .
. . X . . . . . . . . X . .
. . . X . . X . . . . X . .
. . . . X . X X . . X . . .
. . . . . X . X . X . . . .
. . . . . . . X X . . . . .
. . . . . . . . . . . . . .
. . . . . . . . . . . . . .
. . . . . . . . . . . . . .
```

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)



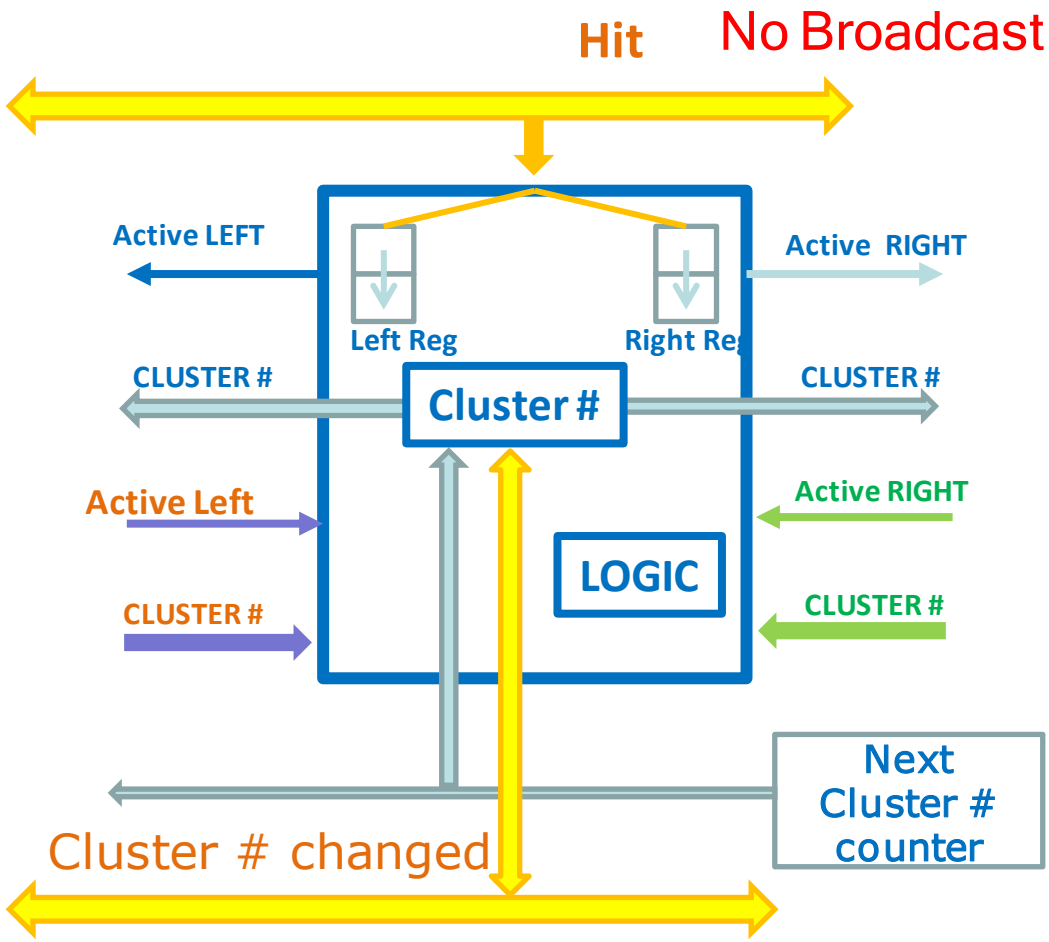

```
. 1 . . . . . . . . . . . . .
. . 1 . . . . . . . . . 2 . .
. . . 1 . . 3 . . . . . 2 . .
. . . . 1 . 3 3 . . 2 . . . .
. . . . . 1 . 3 . 2 . . . . .
. . . . . . . X X . . . . . .
. . . . . . . . . . . . . . .
. . . . . . . . . . . . . . .
. . . . . . . . . . . . . . .
```

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)



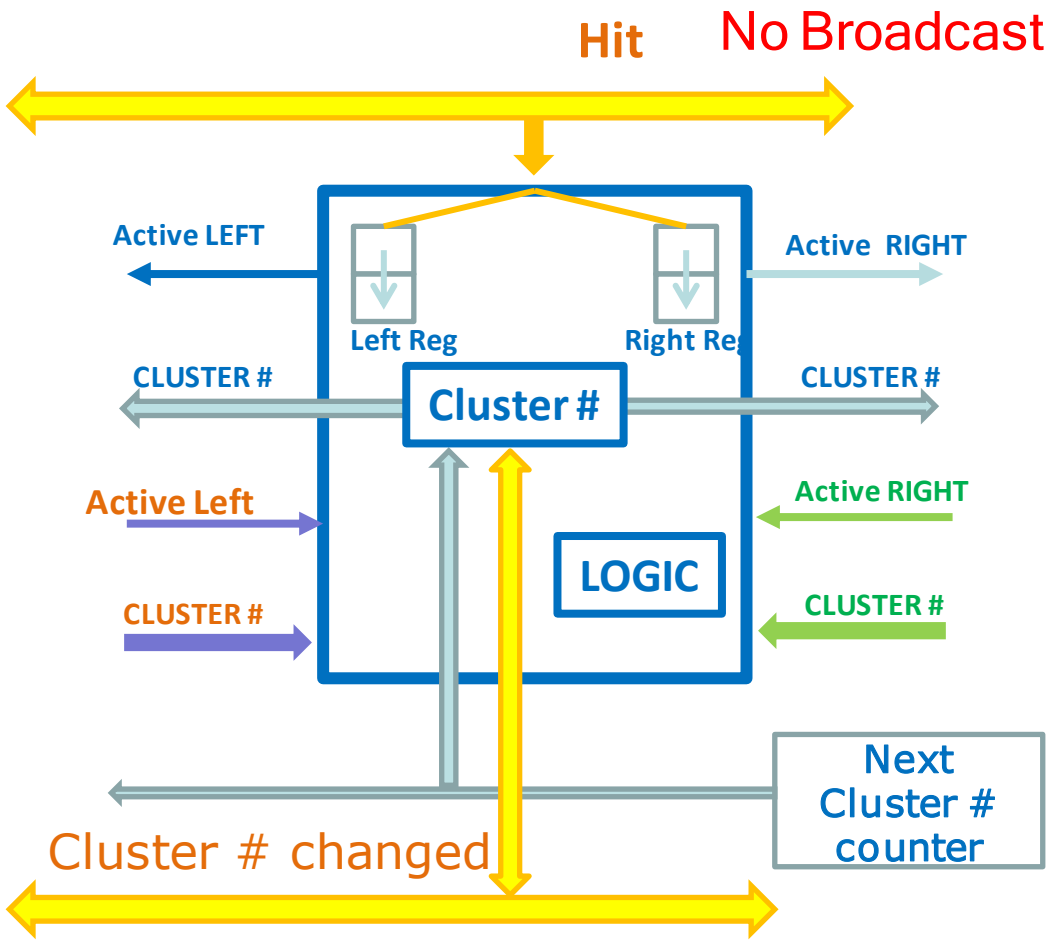
.	1	.	.	.	.	.	.	.	.	.	.	.	.
.	.	1	.	.	.	.	.	.	.	.	2	.	.
.	.	.	1	.	.	3	.	.	.	.	2	.	.
.	.	.	.	1	.	3	3	.	.	2	.	.	.
.	.	.	.	.	1	.	3	.	2	.	.	.	.
.	.	.	.	.	.	.	2	2	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.

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)



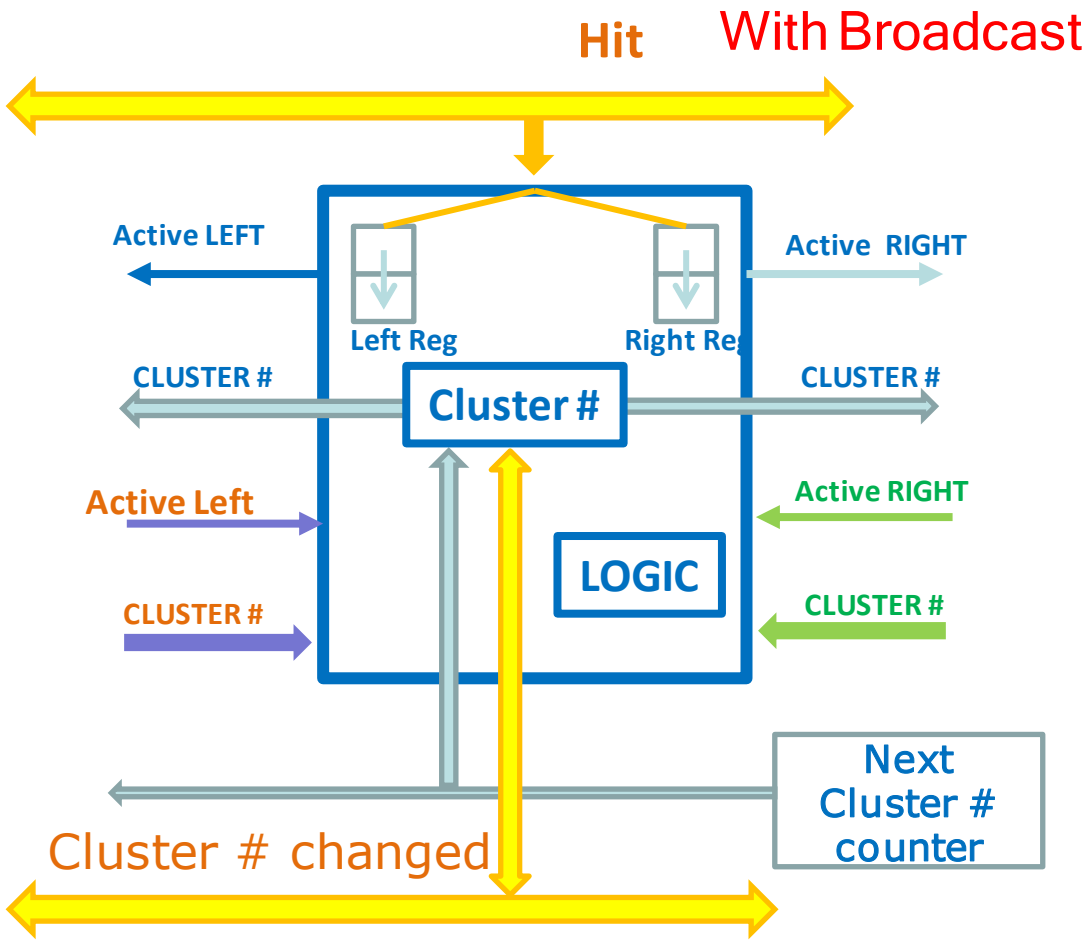
.	1	.	.	.	.	.	.	.	.	.	.	.	.
.	.	1	.	.	.	.	.	.	.	.	2	.	.
.	.	.	1	.	.	2	.	.	.	.	2	.	.
.	.	.	.	1	.	2	2	.	.	2	.	.	.
.	.	.	.	.	1	.	2	.	2	.	.	.	.
.	.	.	.	.	.	.	2	2	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.

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 (5)



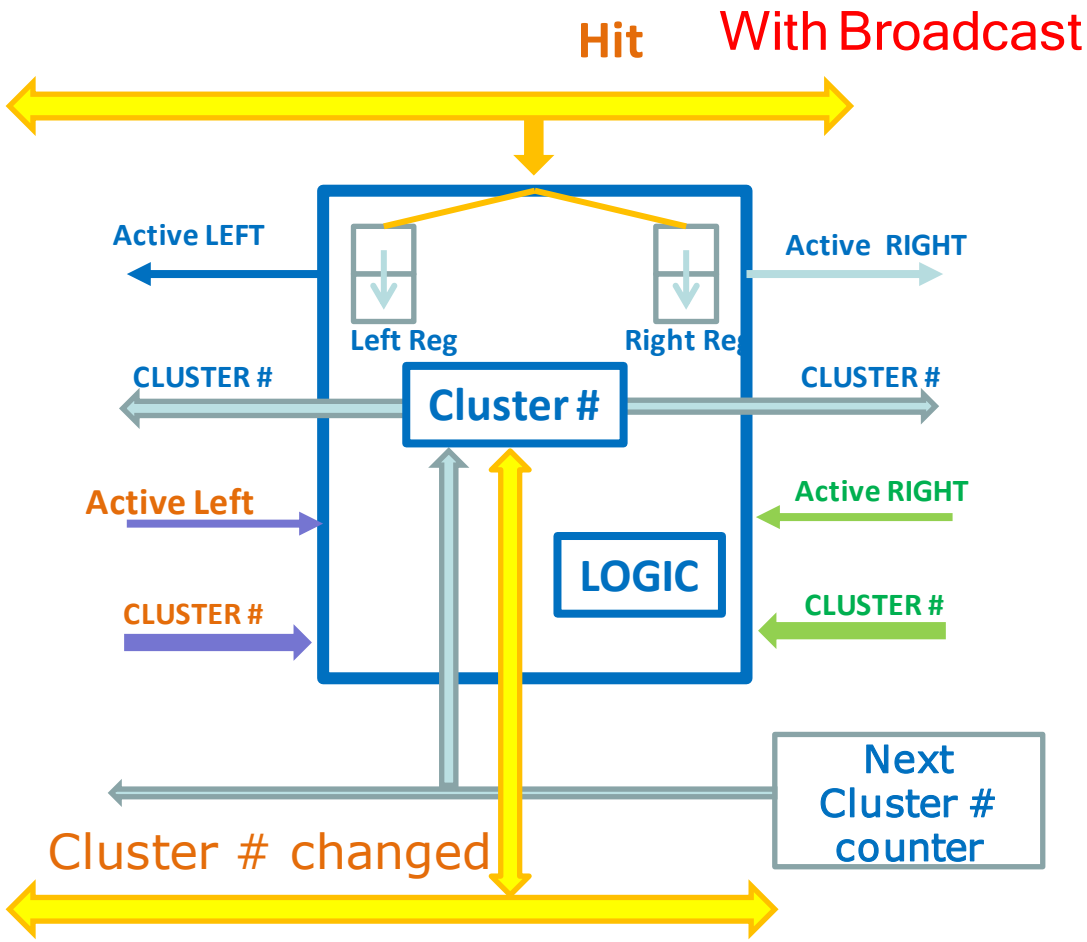
.	1	.	.	.	.	.	.	.	.	.	.	.	.
.	.	1	.	.	.	.	.	.	.	.	2	.	.
.	.	.	1	.	.	3	.	.	.	.	2	.	.
.	.	.	.	1	.	3	3	.	.	2	.	.	.
.	.	.	.	.	1	.	1	.	2	.	.	.	.
.	.	.	.	.	.	.	X	X	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.

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)



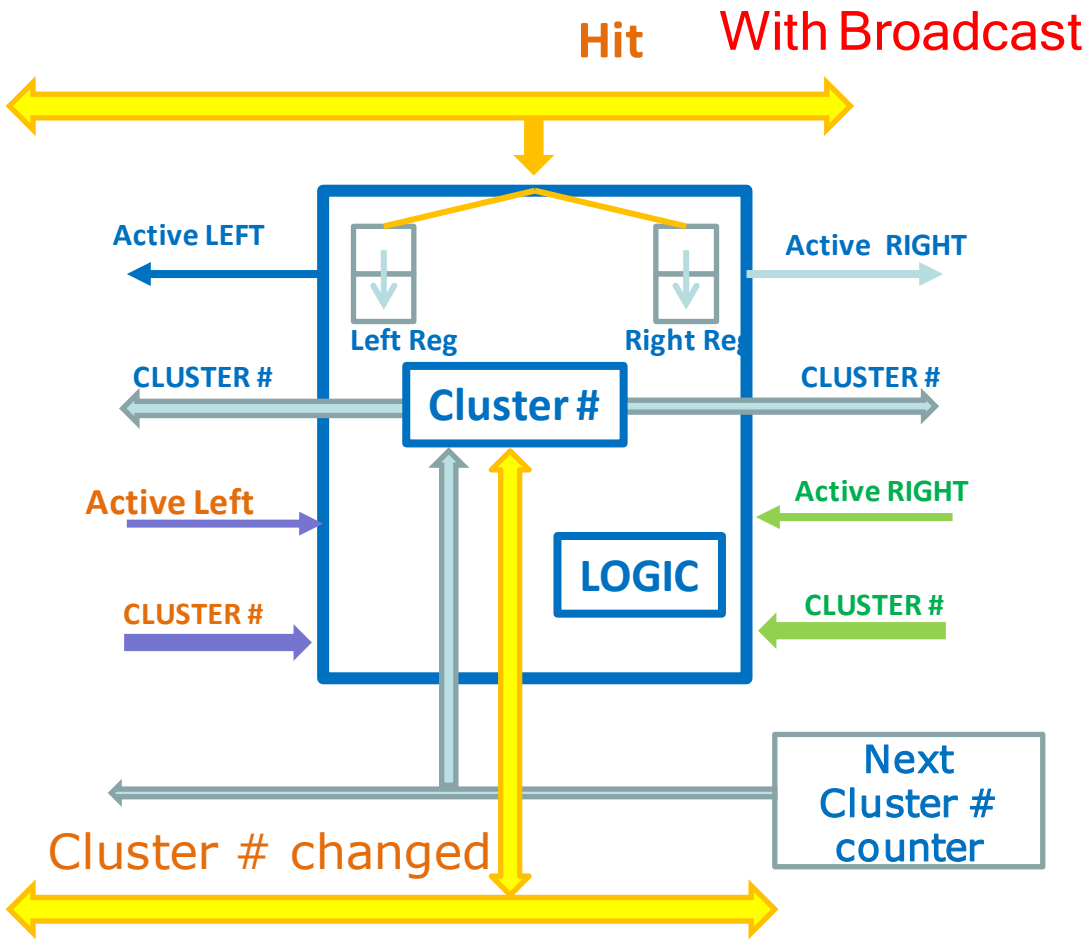
.	1	.	.	.	.	.	.	.	.	.	.	.	.
.	.	1	.	.	.	.	.	.	.	.	2	.	.
.	.	.	1	.	.	3	.	.	.	.	2	.	.
.	.	.	.	1	.	3	3	.	.	2	.	.	.
.	.	.	.	.	1	.	1	.	2	.	.	.	.
.	.	.	.	.	.	.	1	1	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.

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.

Resources and speed

Clustering FSMs 64 columns and 4k clusters:

- 7% of Slices of XC6VLX130T
- 1% of Memory blocks (6 out of 264)
- 100 MHz clock

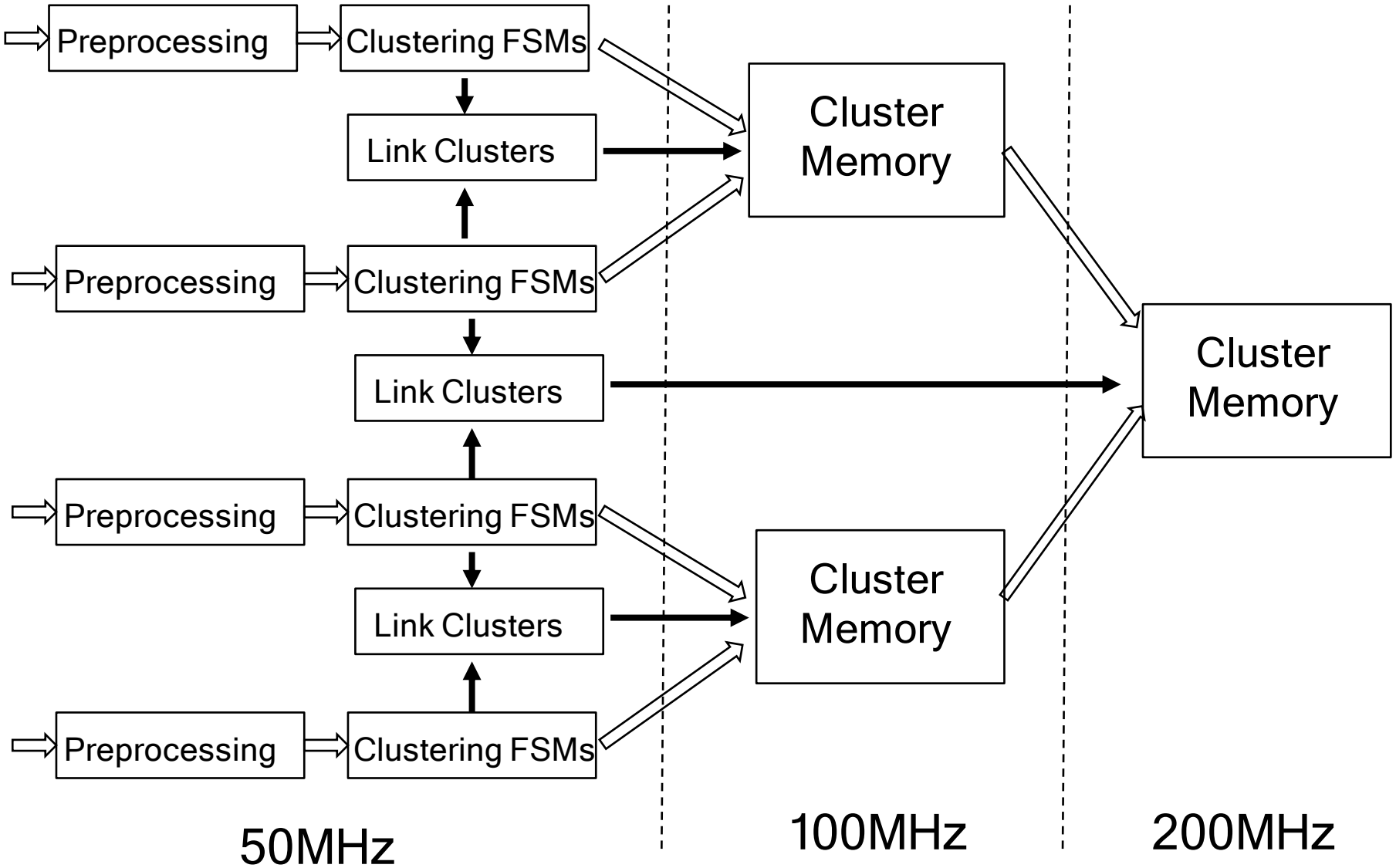
Clustering FSMs 128 columns and 4k clusters:

- 13% of Slices
- 1% of Memory
- 80 MHz

Clustering for 250 columns and 8k clusters:

- 30% of slices
- 30% memory blocks
- 100 MHz

Algorithm data flow



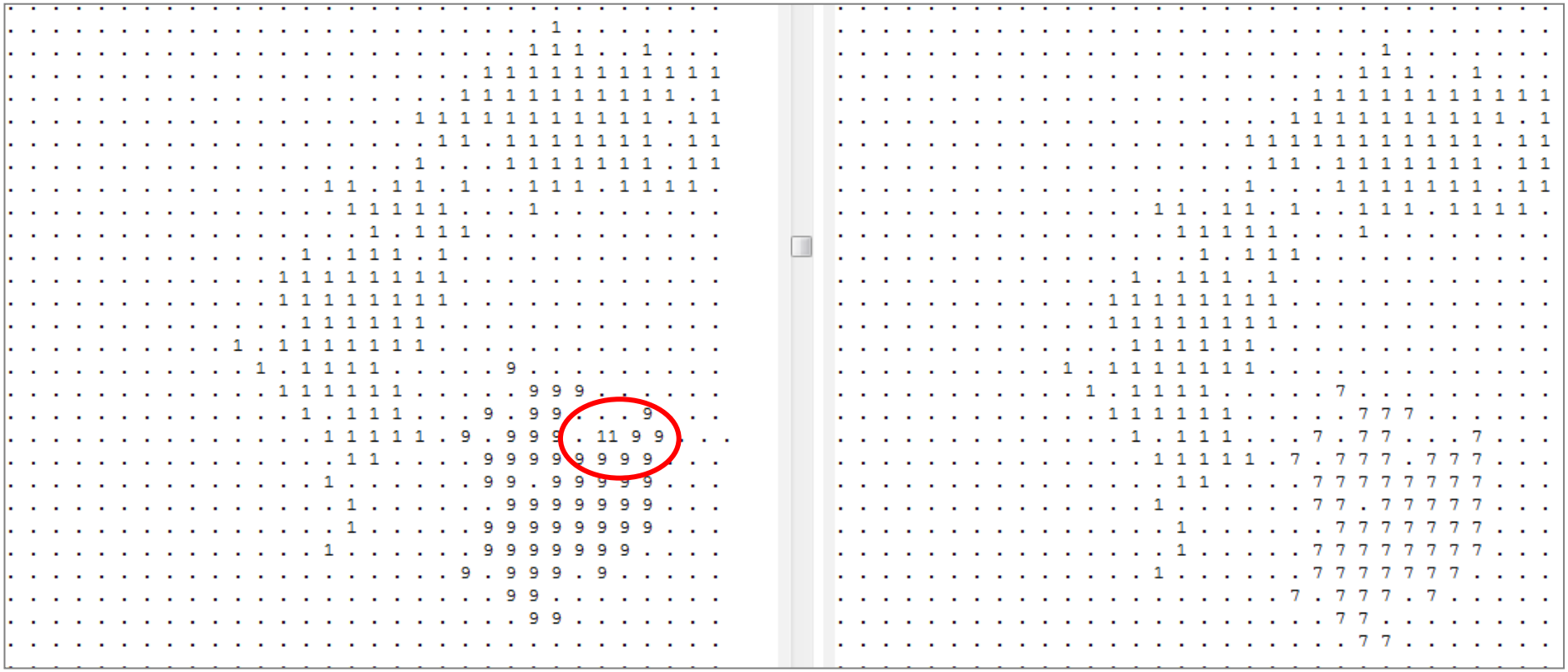
Test in hardware

Clustering algorithm for 64 columns been implemented in DHH and in C++, tested by Martin Gottwald

```
------------------------------------------------------------
Hardware:
. . . . . .
. . . . . .
. . . . . .
. . . . . 1 . . 1 . . . . . .
. . . . . 1 1 1 1 1 . . . . . .
. . . . 1 1 1 1 1 1 1 1 1 . . 1 . 1 1 . . . . . .
. . . . . 1 1 1 1 1 1 1 1 1 1 1 1 . 1 . . . . . .
. . . . . 1 1 1 1 1 1 1 1 1 1 . 1 1 . . . . . .
. . . . . 1 1 1 1 1 1 1 1 1 1 1 . . . . . .
. . . . . 1 1 1 1 1 1 1 1 1 . 1 1 . 1 . . . . . .
. . . . . 1 1 1 1 1 1 1 1 1 1 1 1 1 1 . . . . . .
. . . . . 1 . . 1 . 1 . . 1 1 1 1 1 1 . . . . . .
. . . . . 1 1 1 1 1 1 . . . . . .
. . . . . 1 1 1 1 . . . . . .
. . . . . 1 . 1 1 1 1 . . . . . .
. . . . . 1 1 . . . . . .
. . . . . 1 . . . . . .
. . . . . .
. . . . . .
```

```
Software:
. . . . . .
. . . . . .
. . . . . .
. . . . . 1 . . 1 . . . . . .
. . . . . 1 1 1 1 1 . . . . . .
. . . . 1 1 1 1 1 1 1 1 1 . . 1 . 1 1 . . . . . .
. . . . . 1 1 1 1 1 1 1 1 1 1 1 1 . 1 . . . . . .
. . . . . 1 1 1 1 1 1 1 1 1 1 . 1 1 . . . . . .
. . . . . 1 1 1 1 1 1 1 1 1 1 1 . . . . . .
. . . . . 1 1 1 1 1 1 1 1 1 . 1 1 . 1 . . . . . .
. . . . . 1 1 1 1 1 1 1 1 1 1 1 1 1 1 . . . . . .
. . . . . 1 . . 1 . 1 . . 1 1 1 1 1 1 . . . . . .
. . . . . 1 1 1 1 1 1 . . . . . .
. . . . . 1 1 1 1 . . . . . .
. . . . . 1 . 1 1 1 1 . . . . . .
. . . . . 1 1 . . . . . .
. . . . . 1 . . . . . .
. . . . . .
. . . . . .
```

Test in hardware 2



Somtimes errors observed

Errors occur only in last cluster => relates to Cluster Memory Update process, it is stopped too early

Questions

1. Biggest part of clusters have very simple shape while all complexity of clustering algorithm comes from wird cluster shapes.

   Is it what we need?

2. What is a goal of clustering ?

   Preserving information of low momentum particles ?

   Calculate center of gravity to improve online special resolution and save CPU time later on?

   Is there a definition of BAD or GOOD clusters ?

Conclusions and outlook

- Clustering Core algorithm implemented
- One bug in update of cluster numbers to be fixed
- Speed 100 MHz , four streams in parallel
- Resources required 30% of Logic and 30% of memory blocks of XC6VLX130T

Nest steps

- Implement merging of clusters
- Full test in real hardware
- To be complete by end of July