

Online Unsupervised Spike Sorting Accelerator

Zhewei Jiang, Jamis Johnson
CSEE4840 Embedded System Design



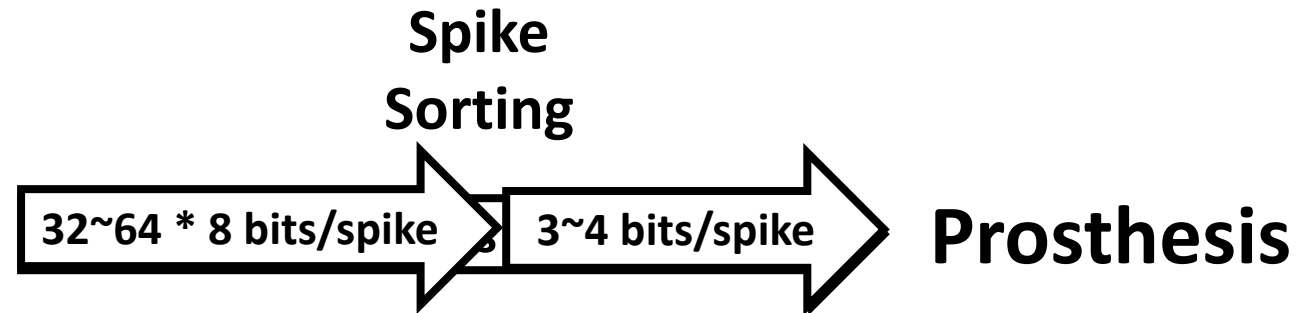
Motivation for Hardware Spike Sorting

- Motivated by Transmission Cost

- High energy consumption
- Throughput limitation

- Challenges

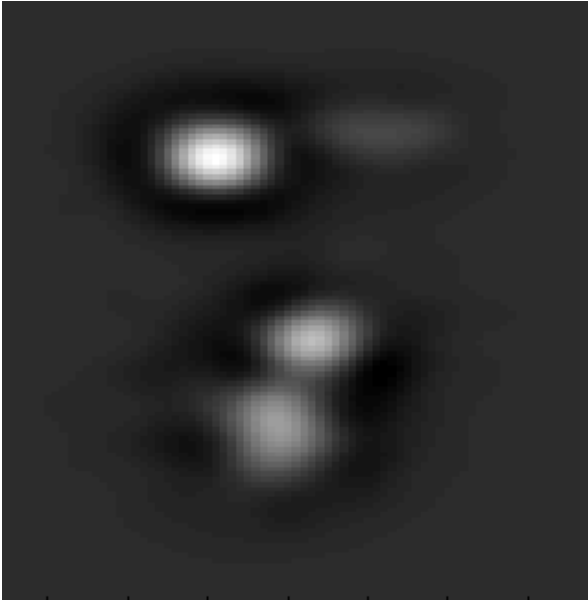
- Computation complexity
 - ❖ Feature extraction
 - ❖ Clustering
- Memory Requirement
 - ❖ Feature Distribution
 - ❖ Cluster Centroid



Algorithm

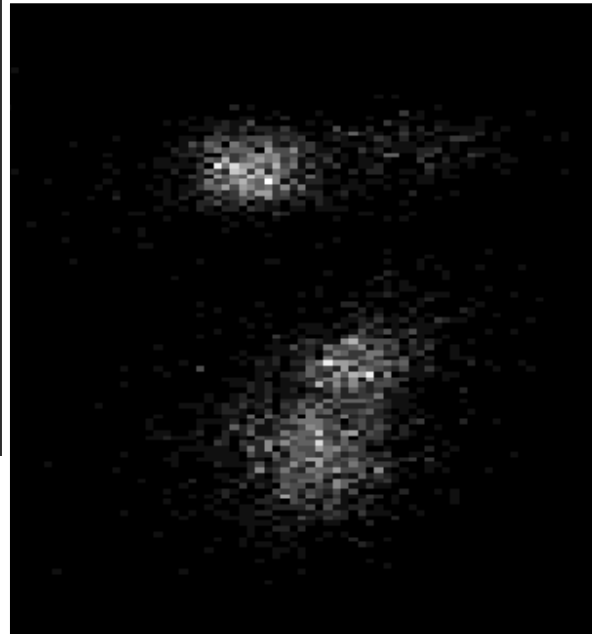
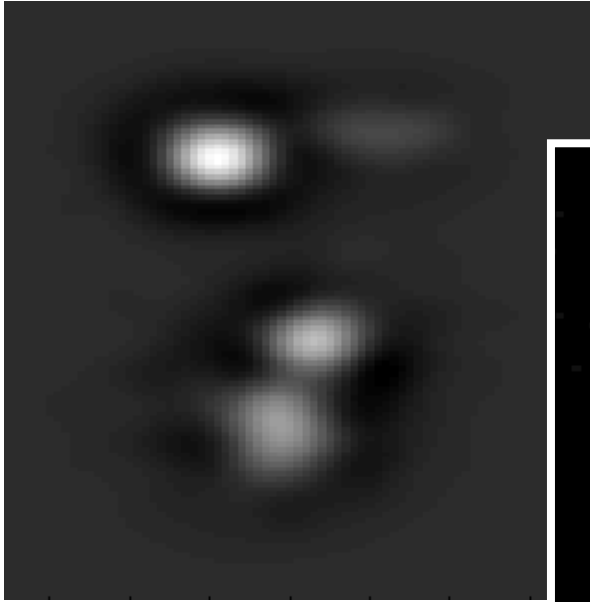
Stage	Algorithm	Main Module(s)
Distribution	1D Kernel Density Estimation	DISTR (main FSM)
Sample Criterion	Laplacian	DISTR (Laplace FSM)
Training	Bimodal prediction & replacement policy	CAM_CLUSTER GRIDFIND
Sorting	Closest Grid (grid granularity)	CAM_CLUSTER GRIDFIND WTA

Algorithm - Kernel Density Estimation



- Approximate Bayesian Optimal
 - Kernel Density Estimation
 - Marr wavelet as kernel
- Hardware cost assuming 8 bit resolution
 - 256^2 memory entry
 - Updates per spike \propto kernel smoothing parameter

Algorithm - Kernel Density Estimation



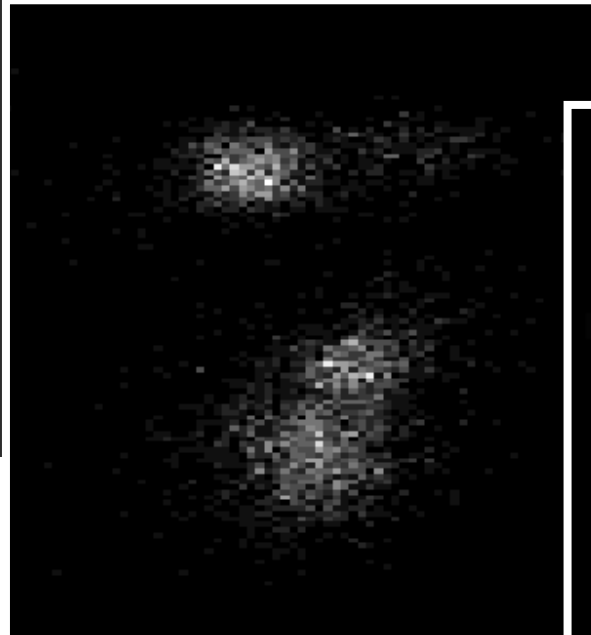
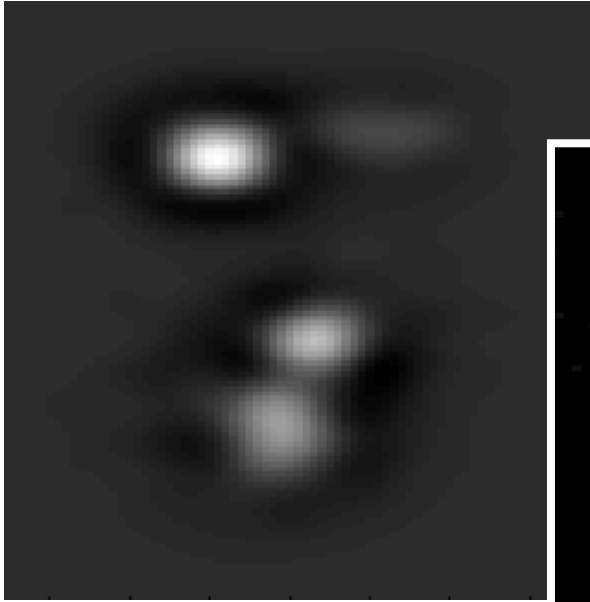
Histogram

- Trivial kernel

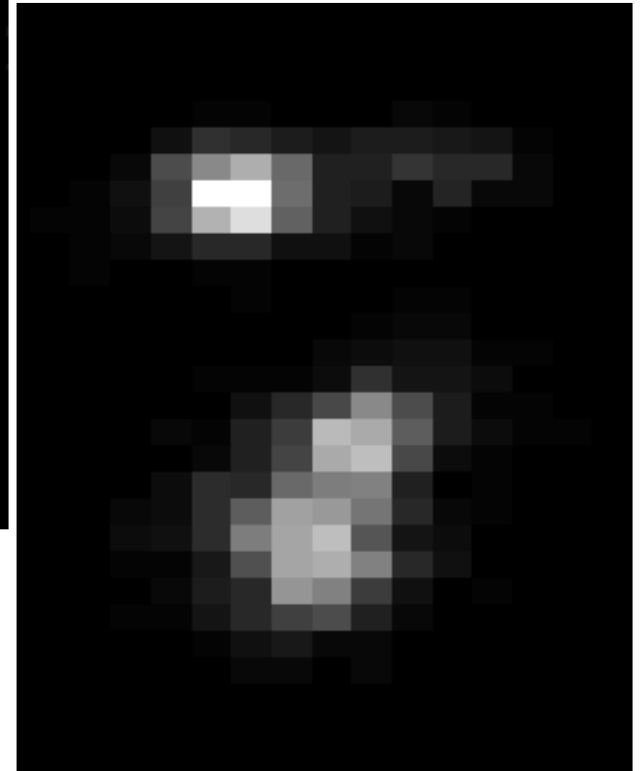
Hardware cost

- 256^2 memory entry
- One update per spike

Algorithm - Kernel Density Estimation

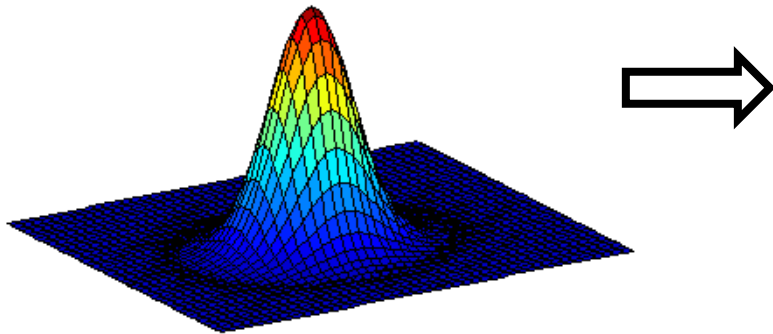


- 64^2 memory entry
- One update per spike



Kernel Density Estimation – Trivial Case

- 2D Kernel Density Estimation
 - Laplacian of Gaussian (Marr Wavelet)



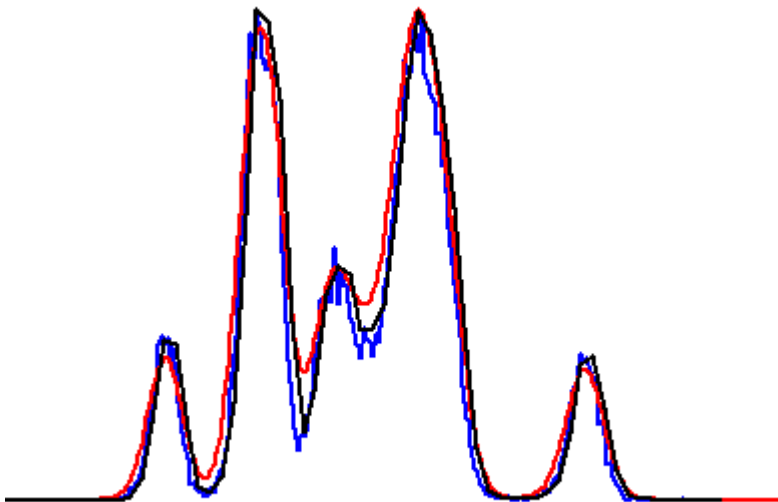
- 1D Kernel Density Estimation
 - Gaussian (or LoG, similar hardware cost if precomputed)

$$kernel(x) = \frac{1}{(\sigma\sqrt{2\pi})^d} * e\left(\frac{-x^2}{2\sigma^2}\right)$$



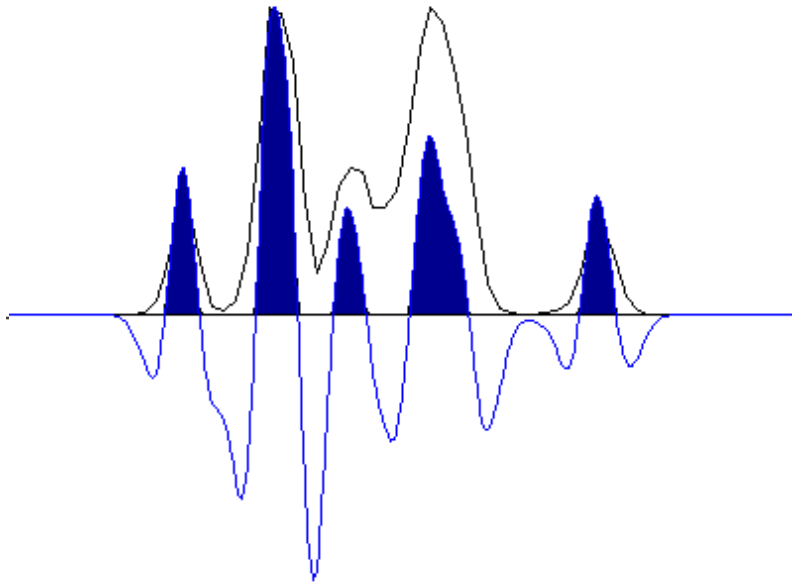
- Histogram
 - Smoothing with bin width

$$Kernel(x, w) = \begin{cases} 1, & w * \left\lfloor \frac{x}{w} \right\rfloor < x < w * \left\lceil \frac{x}{w} \right\rceil \\ 0, & \text{otherwise} \end{cases}$$



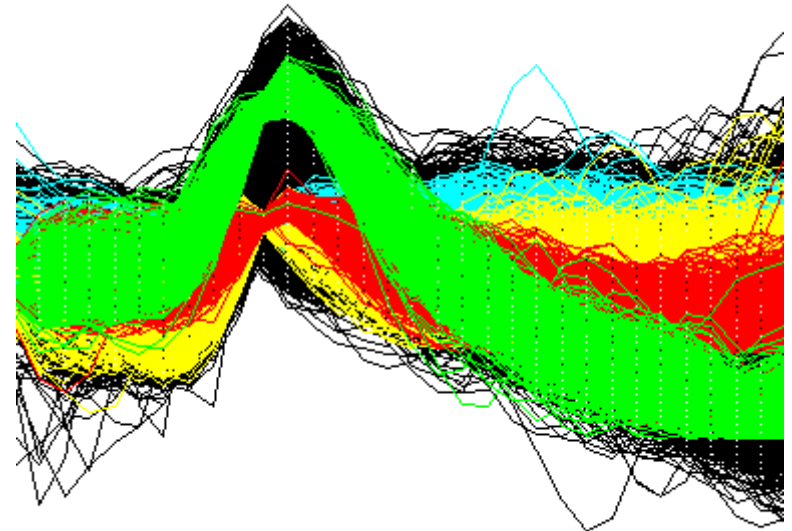
Laplacian

Convolution of histogram of negative Laplace operator $[-1, 2, -1]$



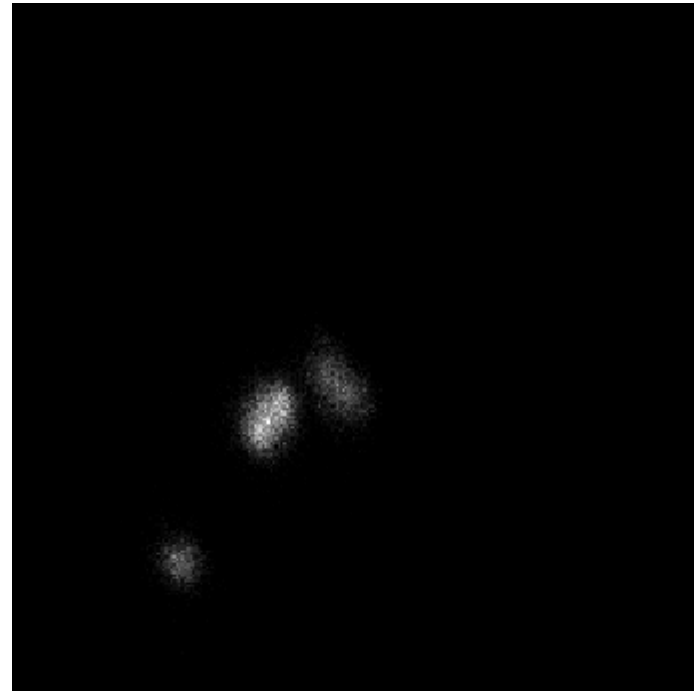
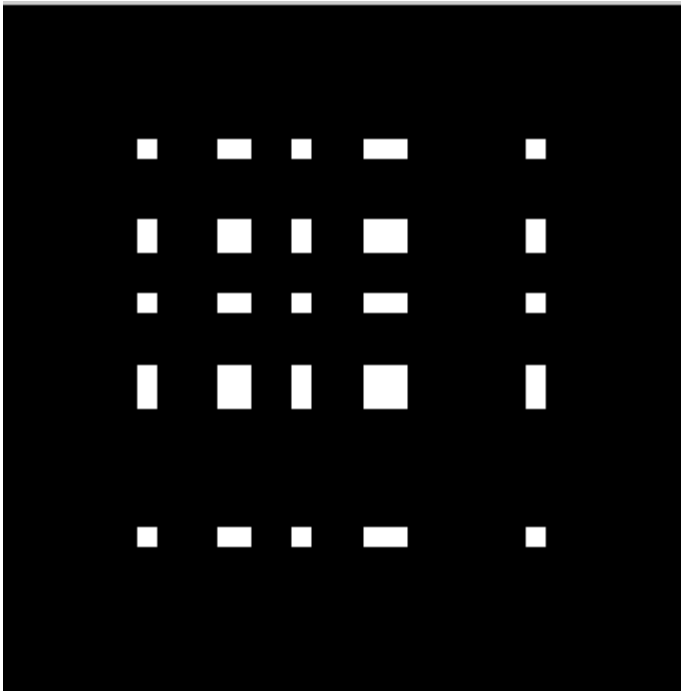
- Only data in the shaded area are used for training

- Downward trend of 2nd derivative is used to segment distribution space into informative and uninformative regions



Informative Sample Vector Mask

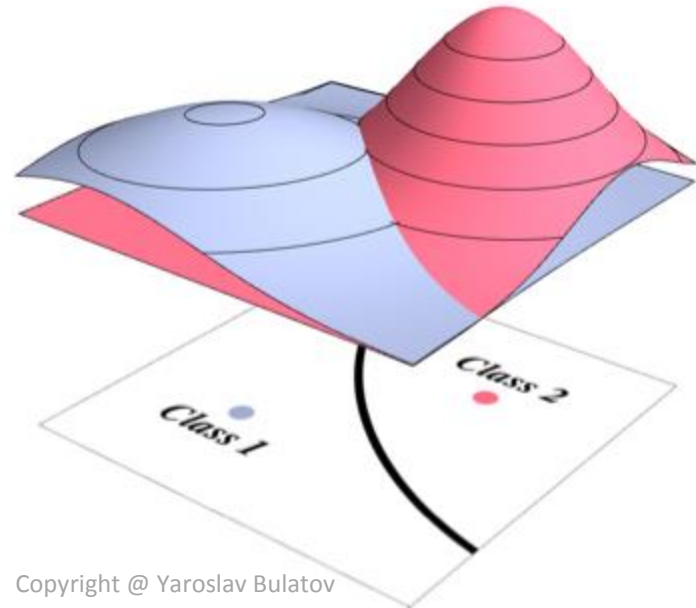
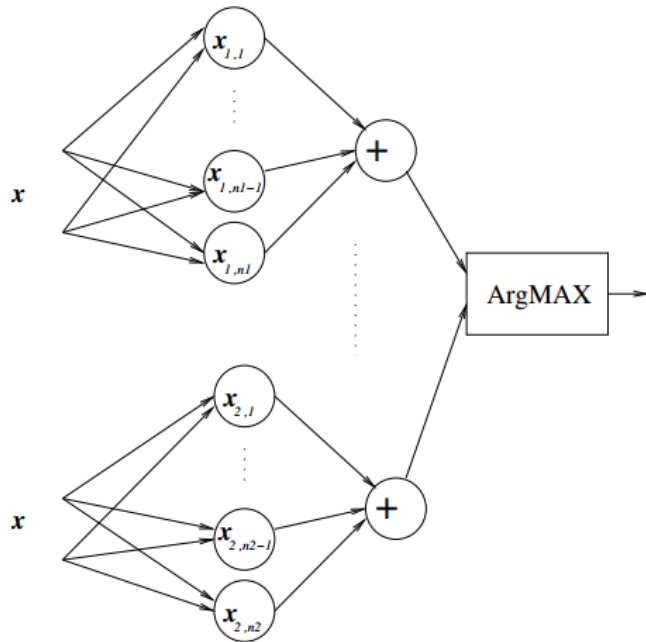
- Informative sample vector mask reduces outliers sensitivity during training phase
- Certain features can share distribution space if no overlap can occur (e.g. peak & hyperpolarization)



Bayesian Classification

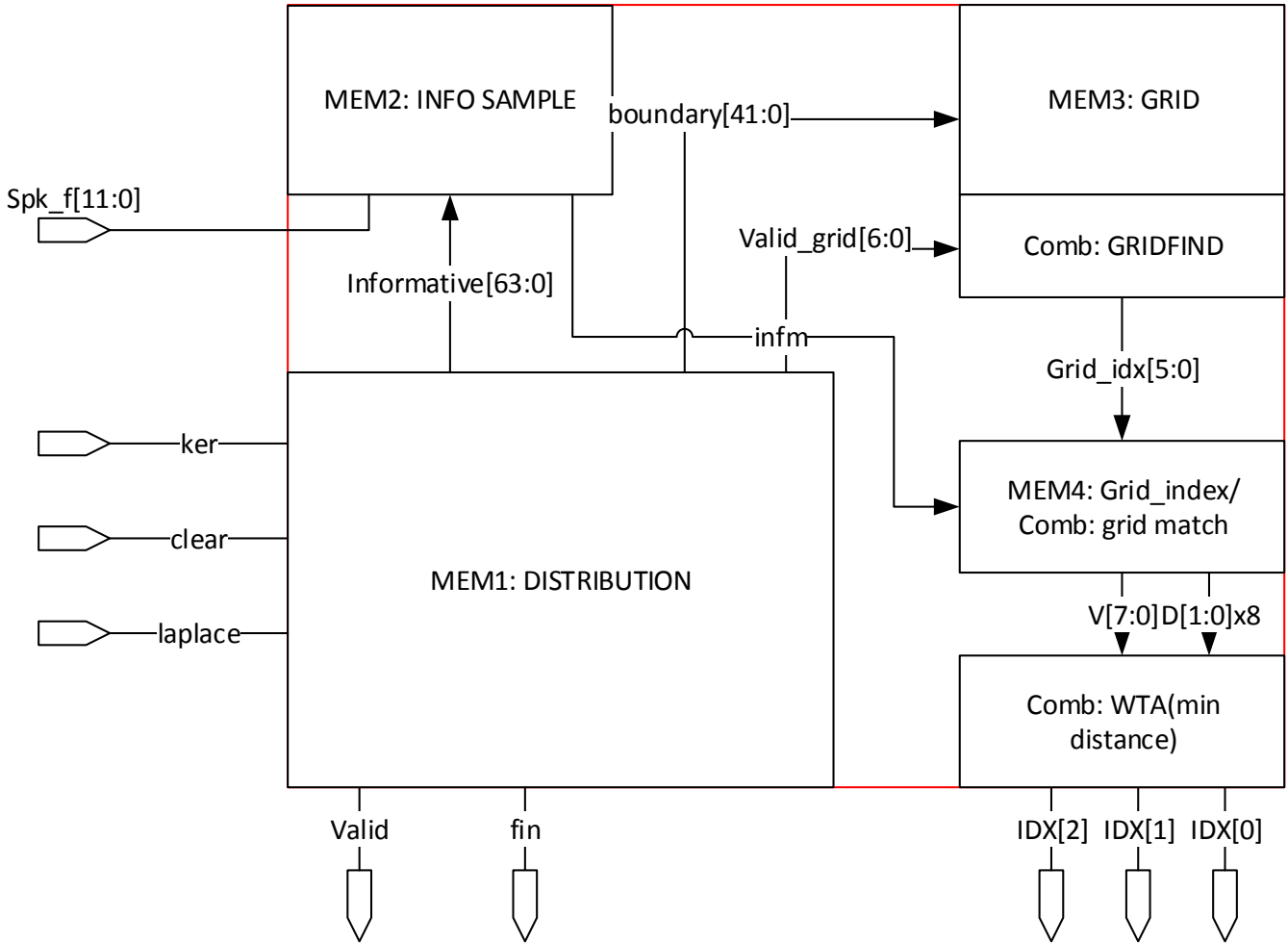
At this point, classification can be done through

- Distance metric (e.g. Probabilistic Neural Network, MCK, ...)
- Boundary metric (Bayesian Boundary)

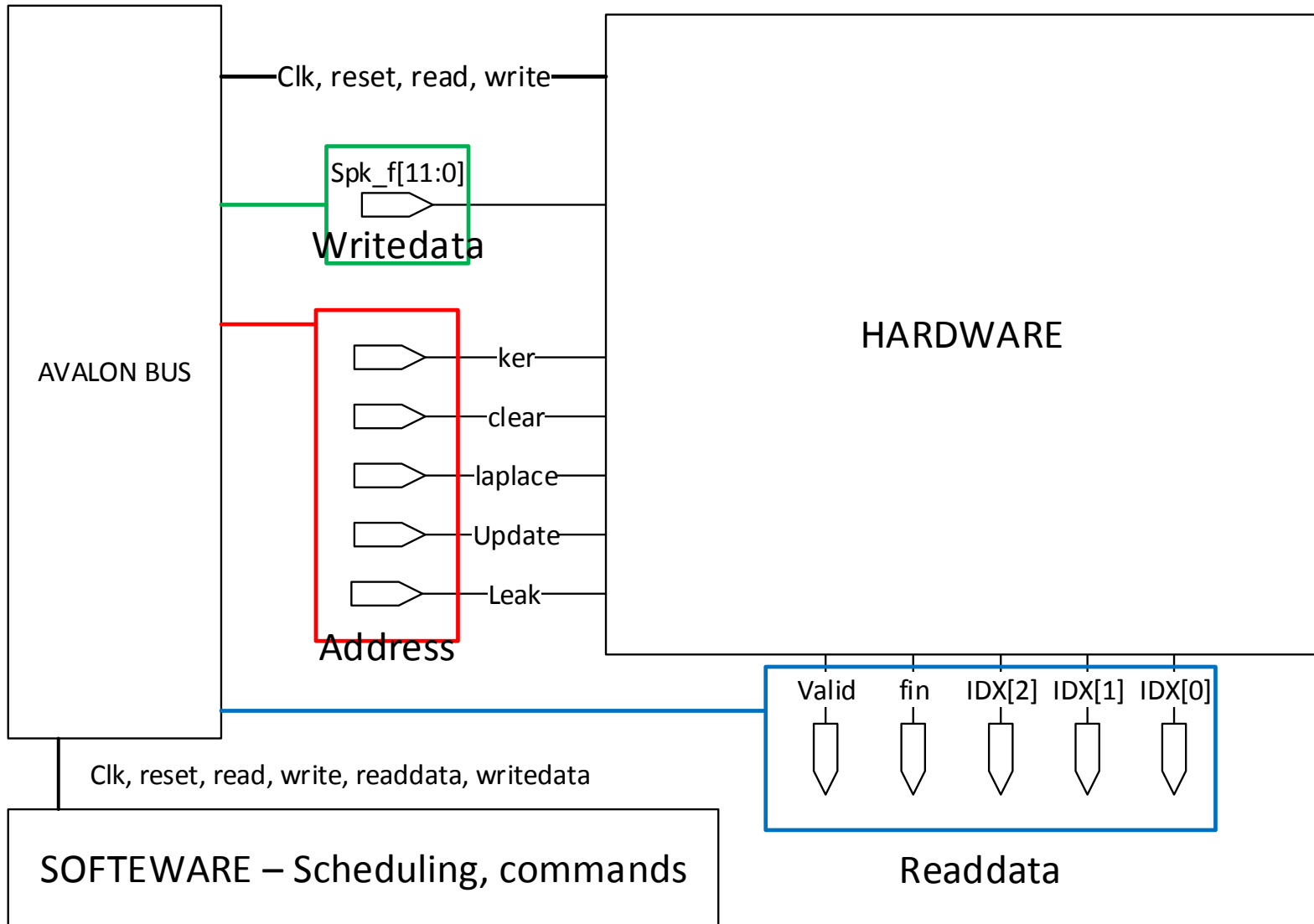


Copyright @ Yaroslav Bulatov

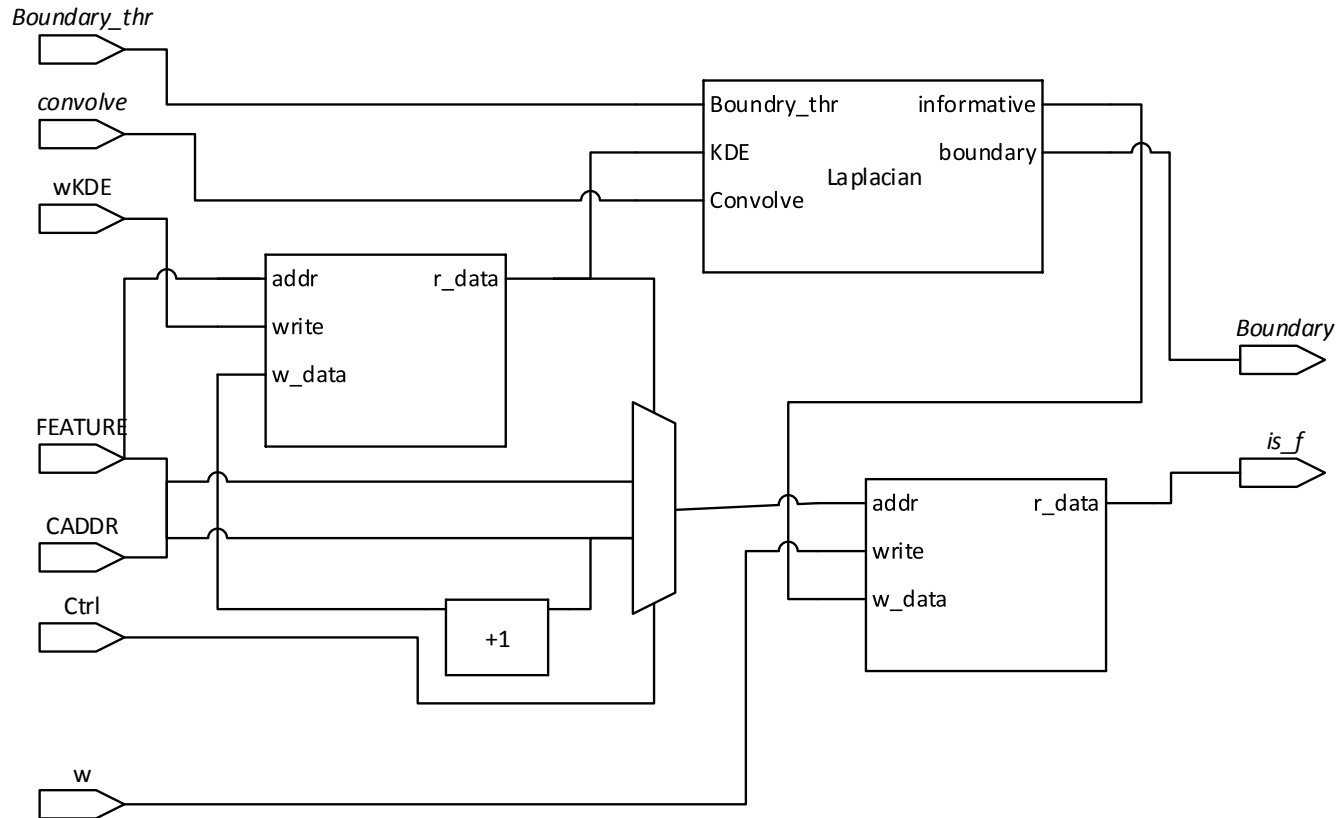
Top-Level Hardware Architecture



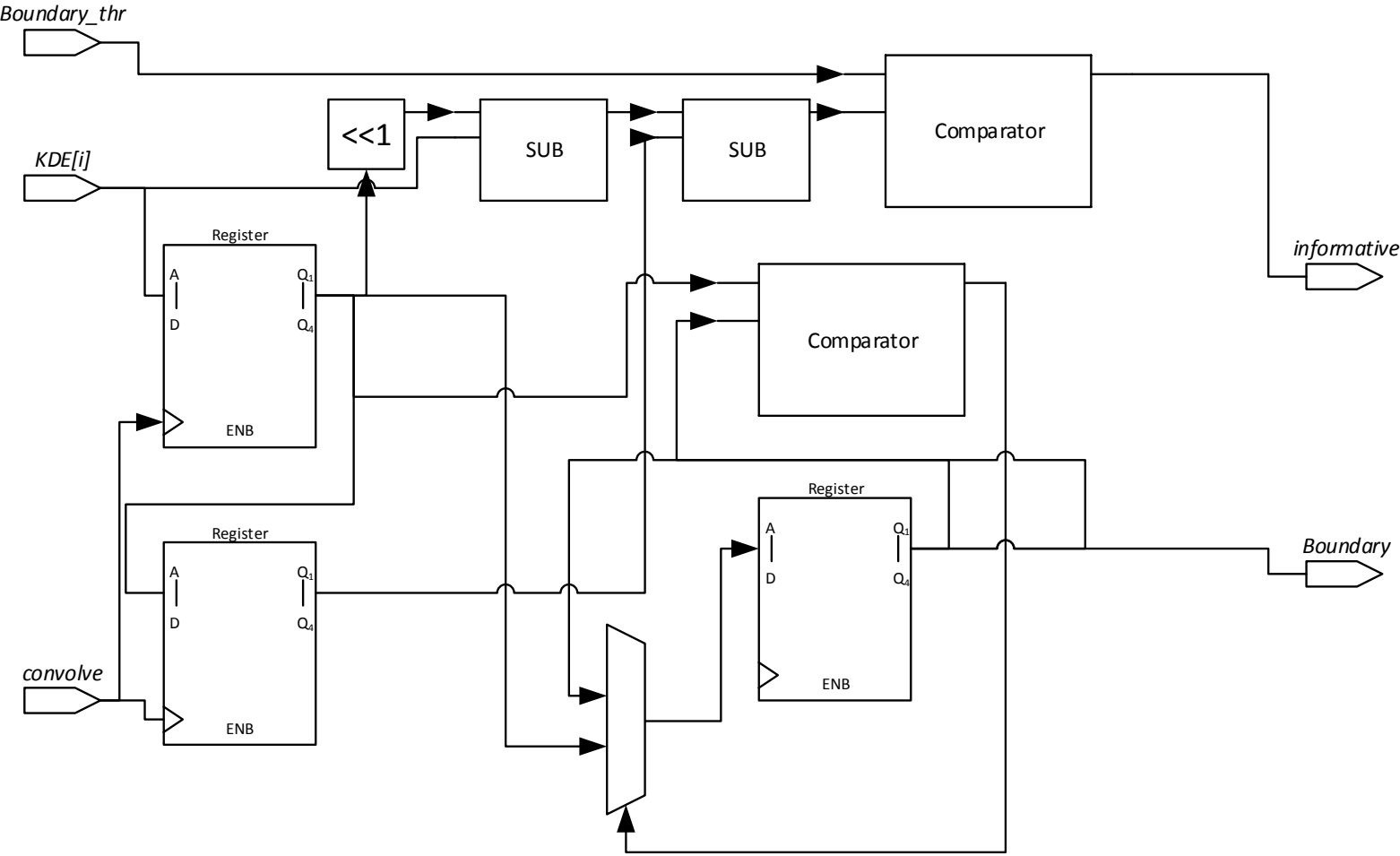
S/H Interface



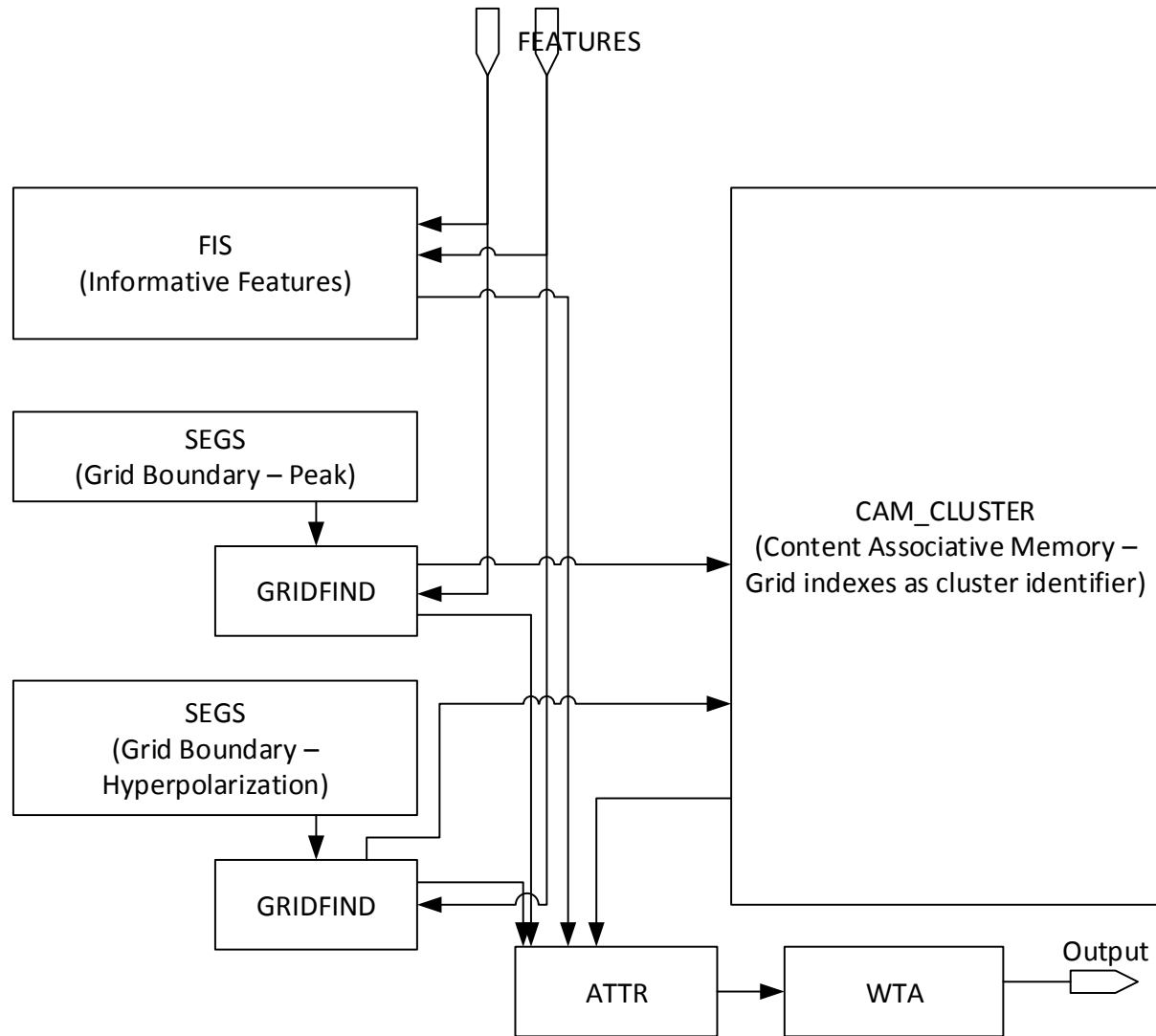
Modules [Distribution]



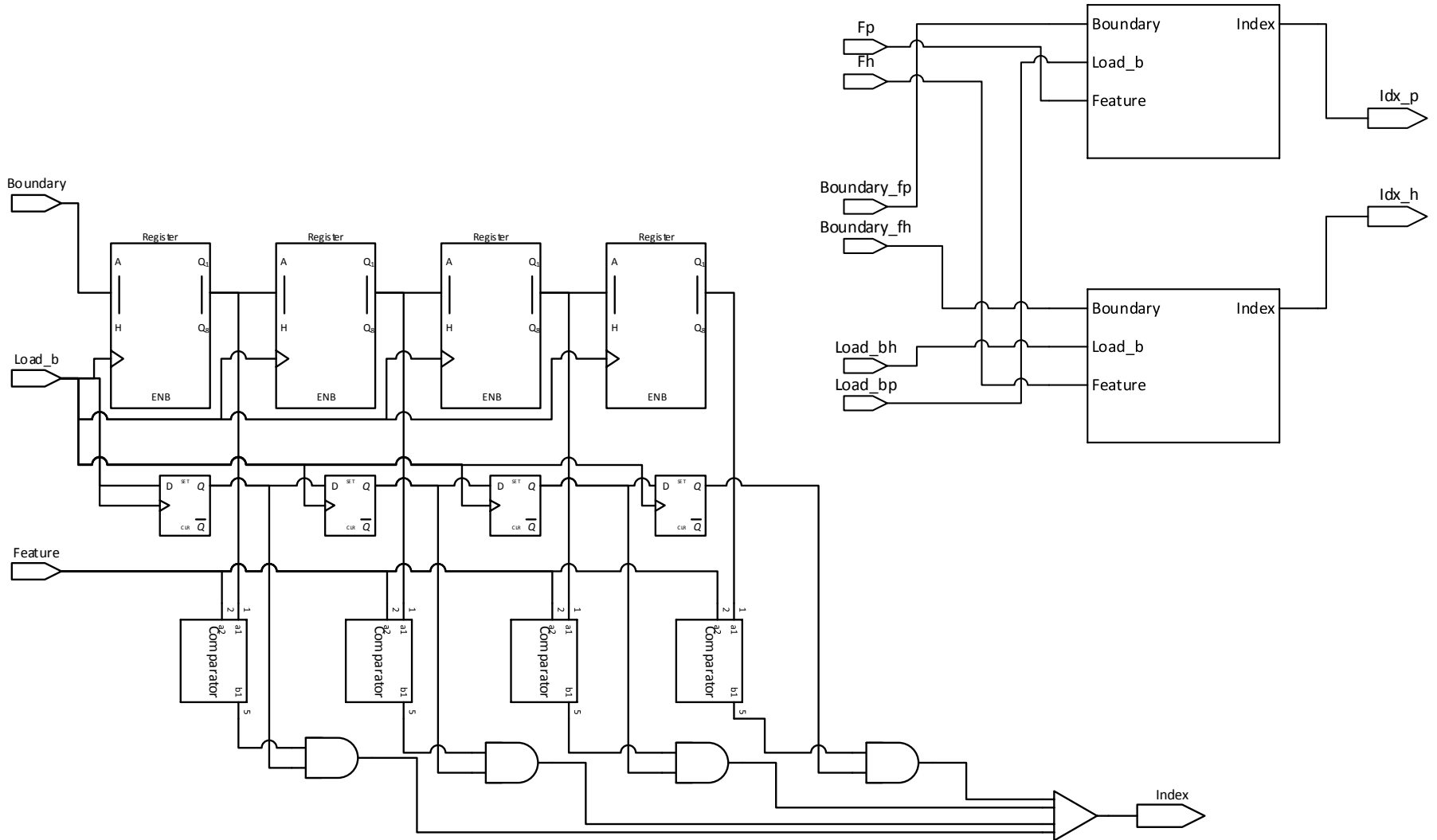
Modules [Distribution]



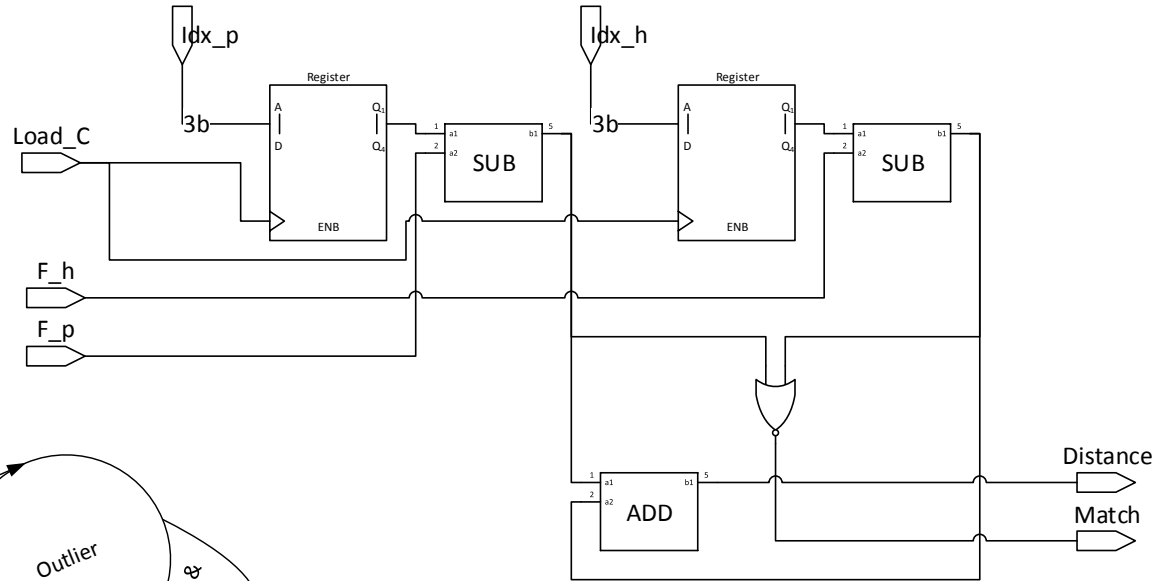
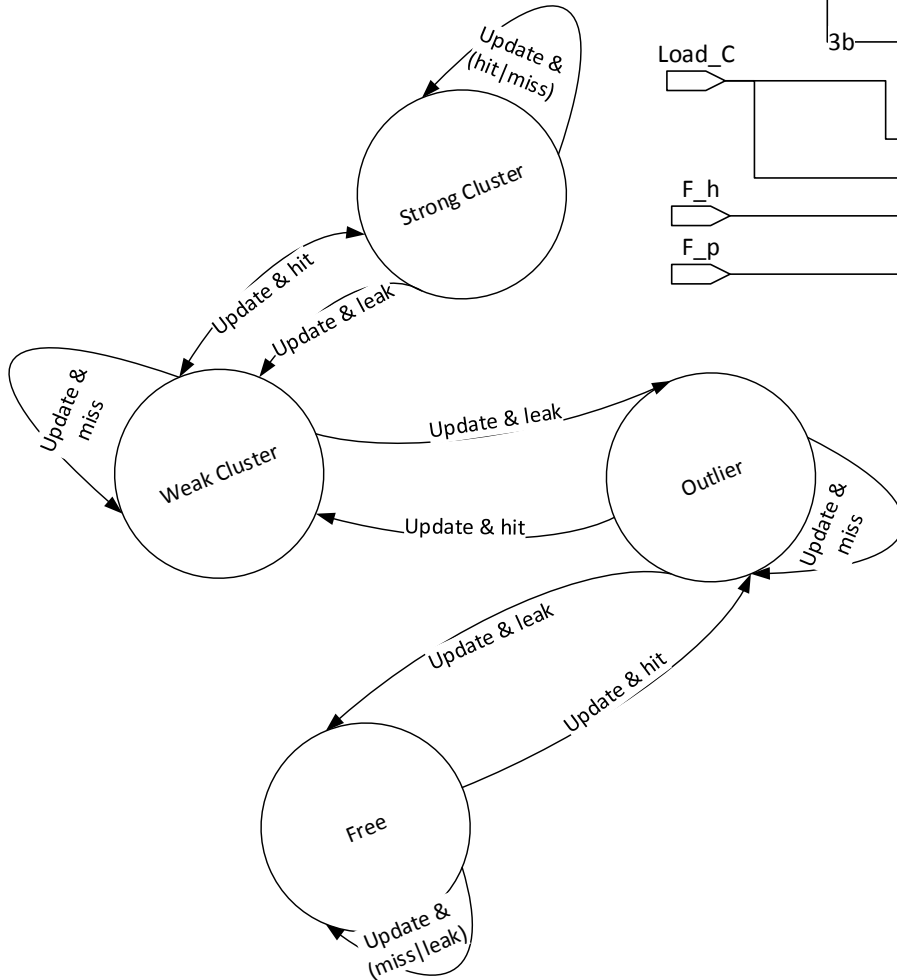
Modules [Training & Sorting]



Modules [GRIDFIND]

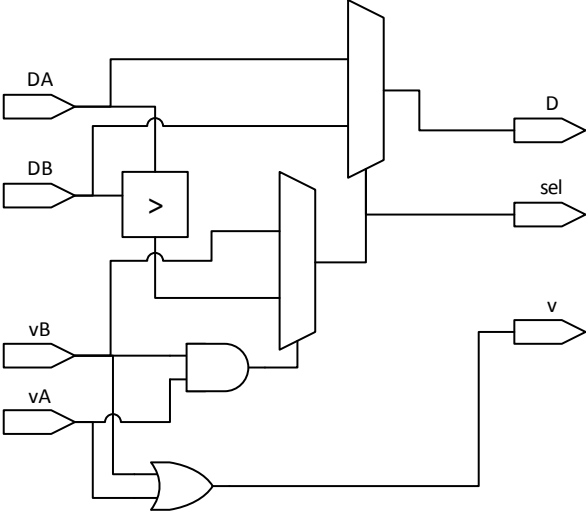
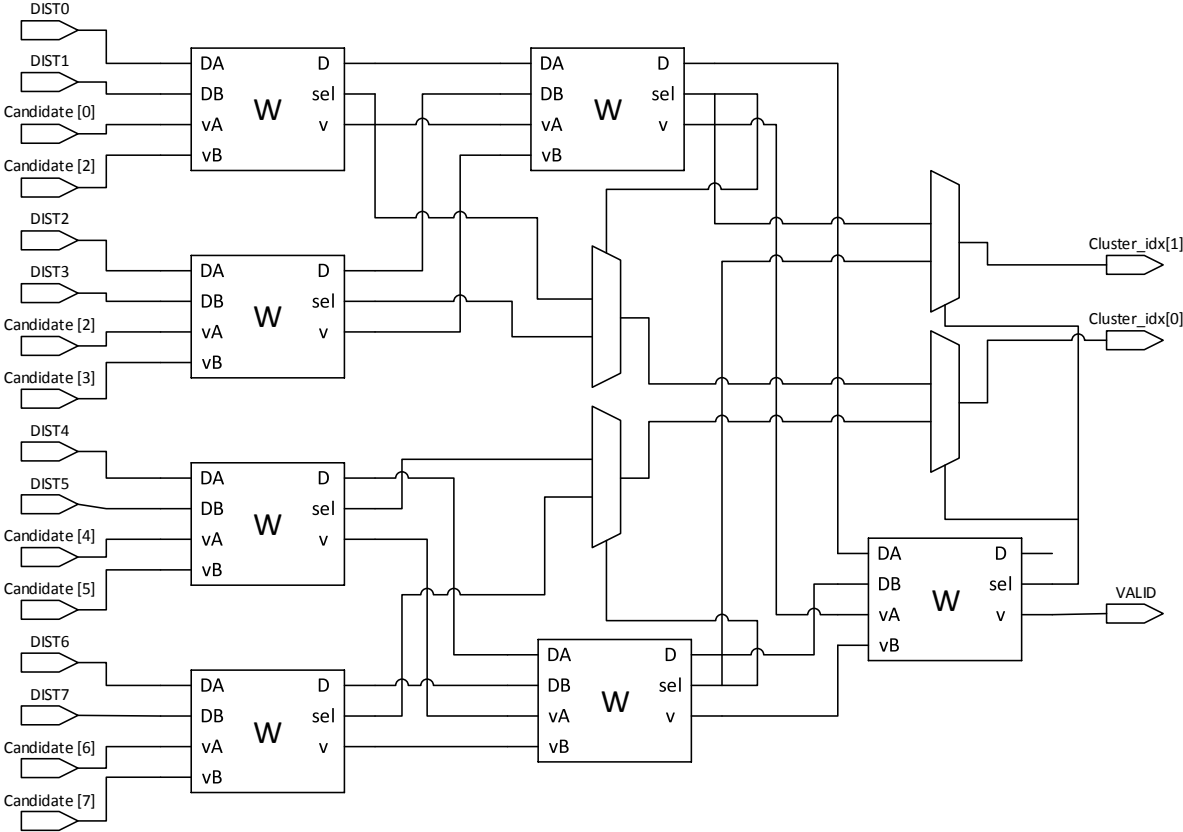


Modules [CAM_CLUSTER]



Each cluster is associated with a bimodal updater, every matching spike strengthens the usefulness of the cluster, and periodic leakage is applied to weaken the cluster. False cluster caused by outliers is cleared this way.

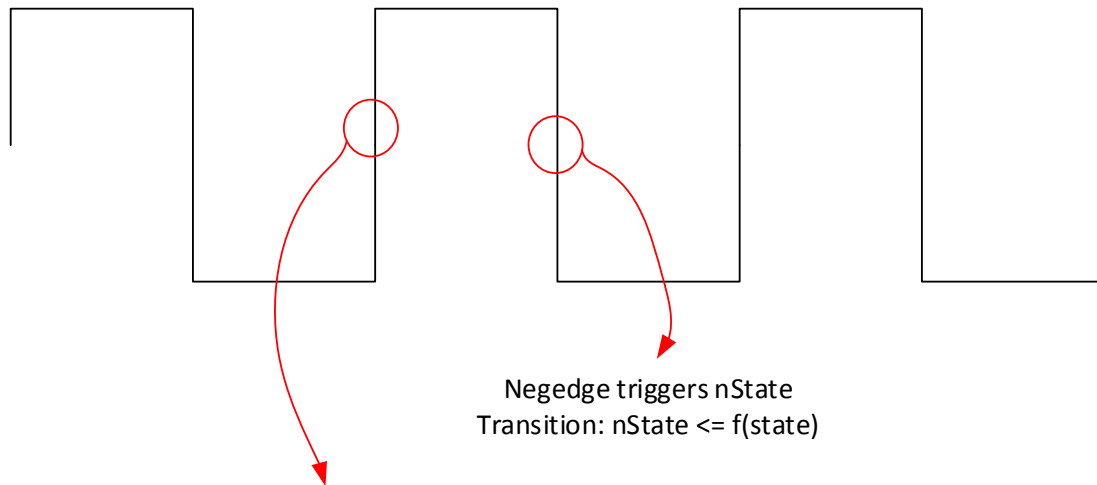
Modules [WTA]



Design – Timing

Sequential Components:

- FSM (for histogram generation in DISTR)
- FSM (for Laplacian in DISTR)
- Usefulness Updater (Bimodal SM for outlier handling in CAM_CLUSTER)

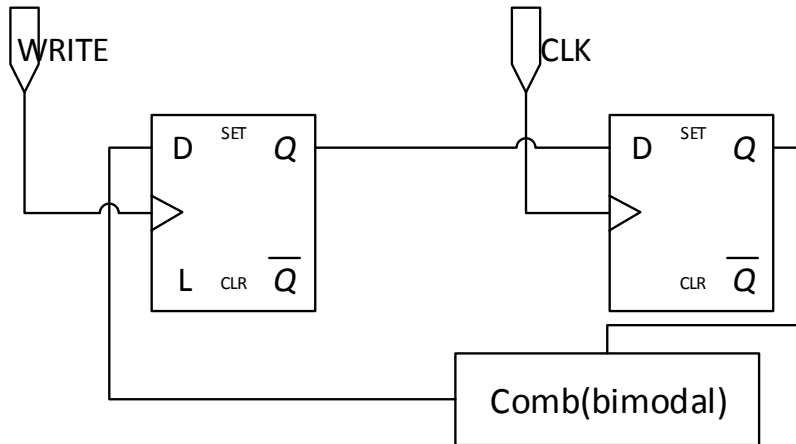
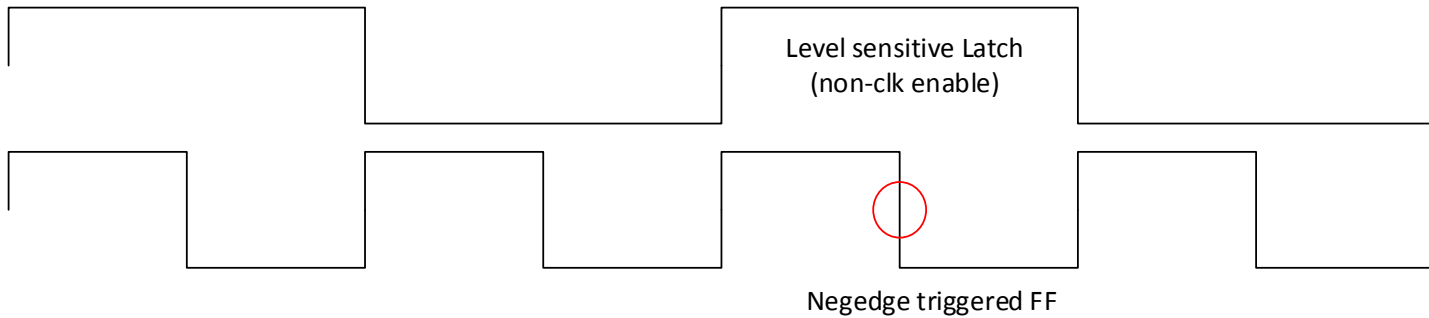


Posedge triggers State Transition: State <= nState
& FSM Outputs (registered output) based on nState

Negedge triggers nState
Transition: nState <= f(state)

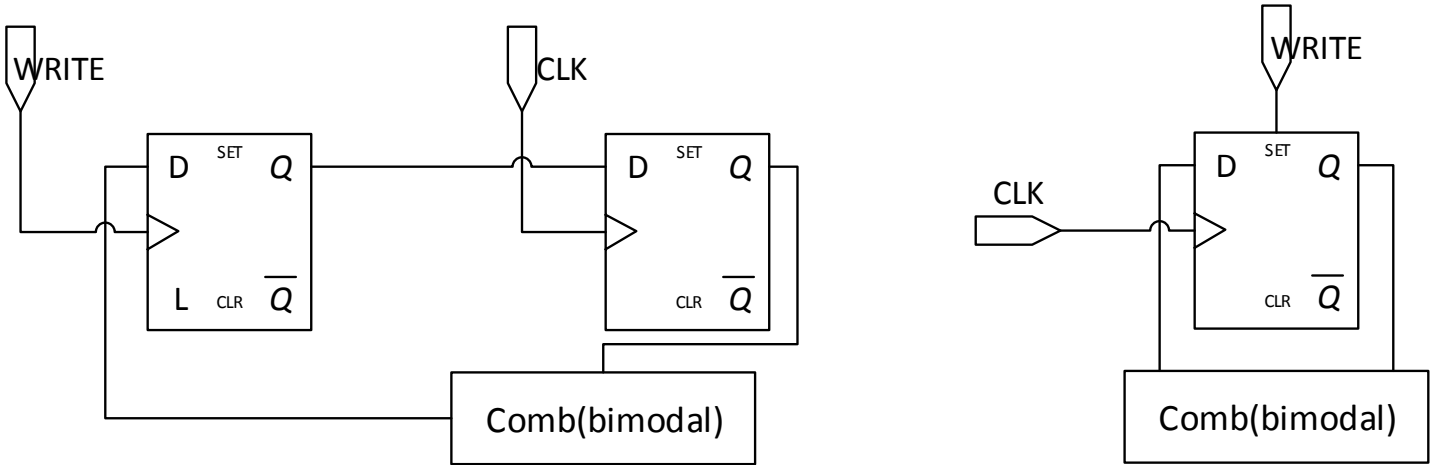
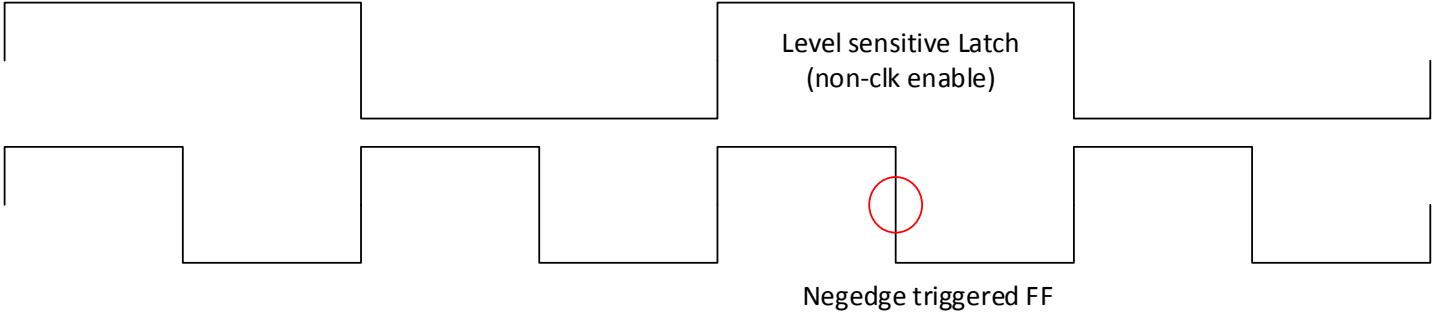
Combinational circuit (in FSM)
propagation for half cycle, guarantee
no set-up/hold violation from clock
jitter or skew

Design – Timing (Legacy)



Retention Latch is used in a power-gating version of the design, with no retention flip-flop available. A non-retention FF trails a Latch to allow the design of a FSM that does not require extra cycle to read back state from L after power gating

Design – Timing (Legacy)



Design – Timing of Module DISTR

Internal FSM timing as previous slides

Ack (signal Fin) based interface, communication locked until software read out “Fin”.

Command Ker: [read Max & incr][write][read Min & incr][write][Fin]

special case when memory overflow:

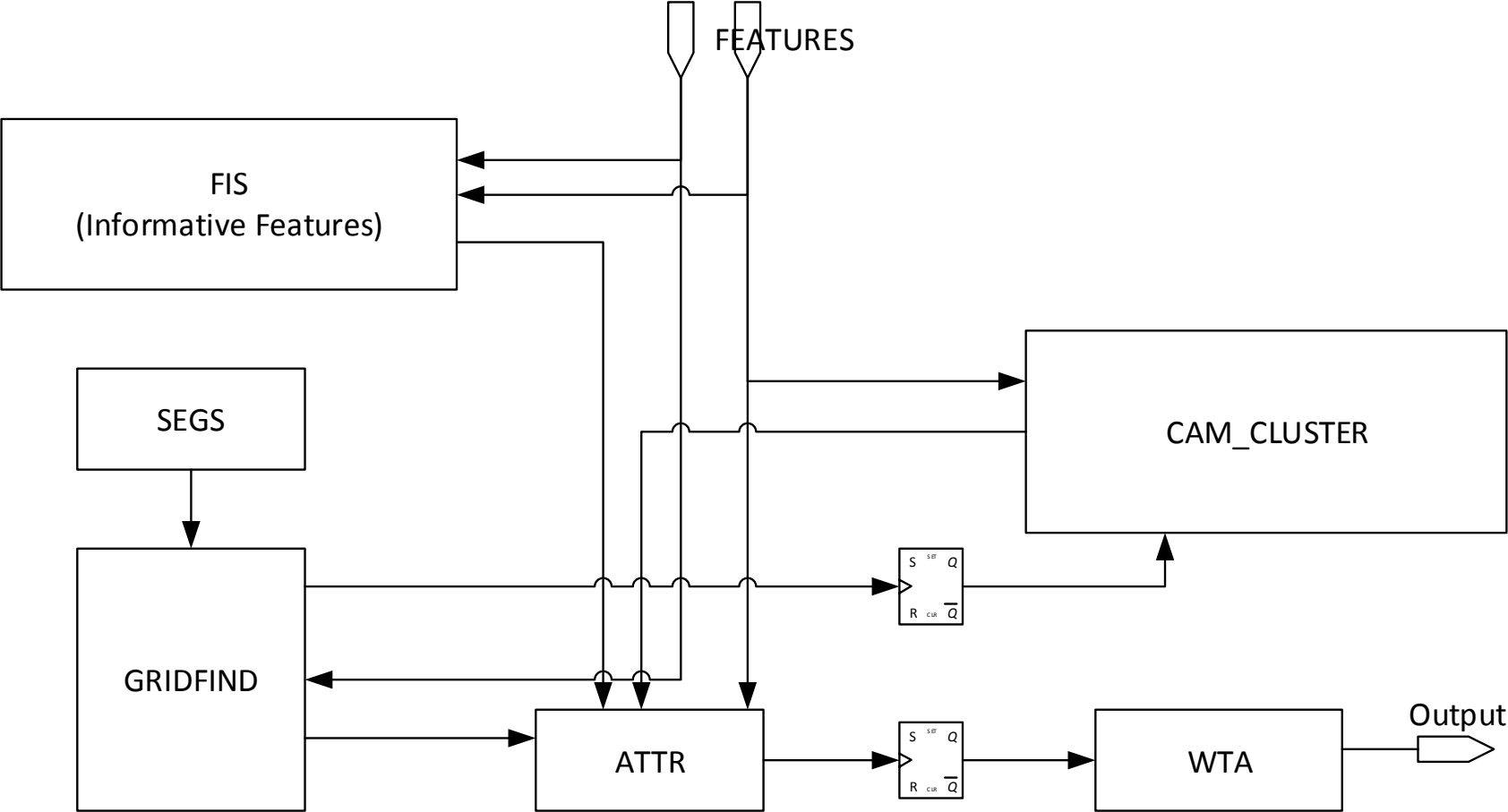
[read & >> 1][write]...(64 entries)

Command Laplace: [read][read][read]...(64 entries)...[Fin]

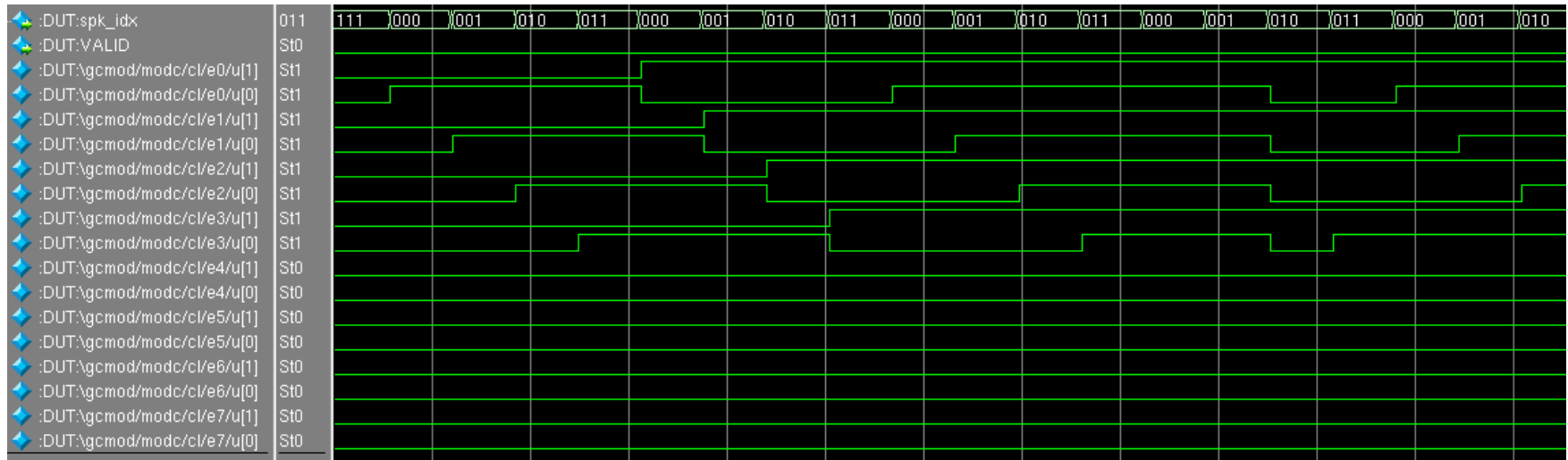
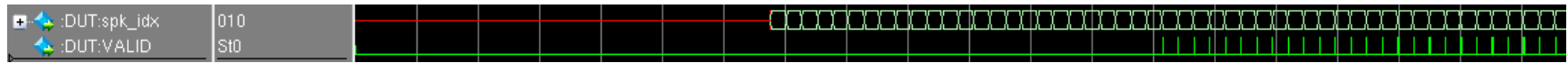
Secondary FSM find informative sample regions based on

$(\text{entry}[i] \ll 1) > (\text{entry}[i-1] + \text{entry}[i+1])$

Design – Timing of Module GC



RTL Sim (Design Compiler)



Issues & Lessons Learned

- Testability
 - Individual Modules
 - Overall Functionality
- Debugging