CMS Phase-II Trigger Upgrade

Sudeshna Banerjee

March 8, 2018

CMS Trigger system

(current status)

Trigger: Choose the interesting events -> top quark, Higgs boson, exotic particles

Triggering is performed in two steps: Level – 1 (Hardware based)
High Level Trigger (software algorithms)

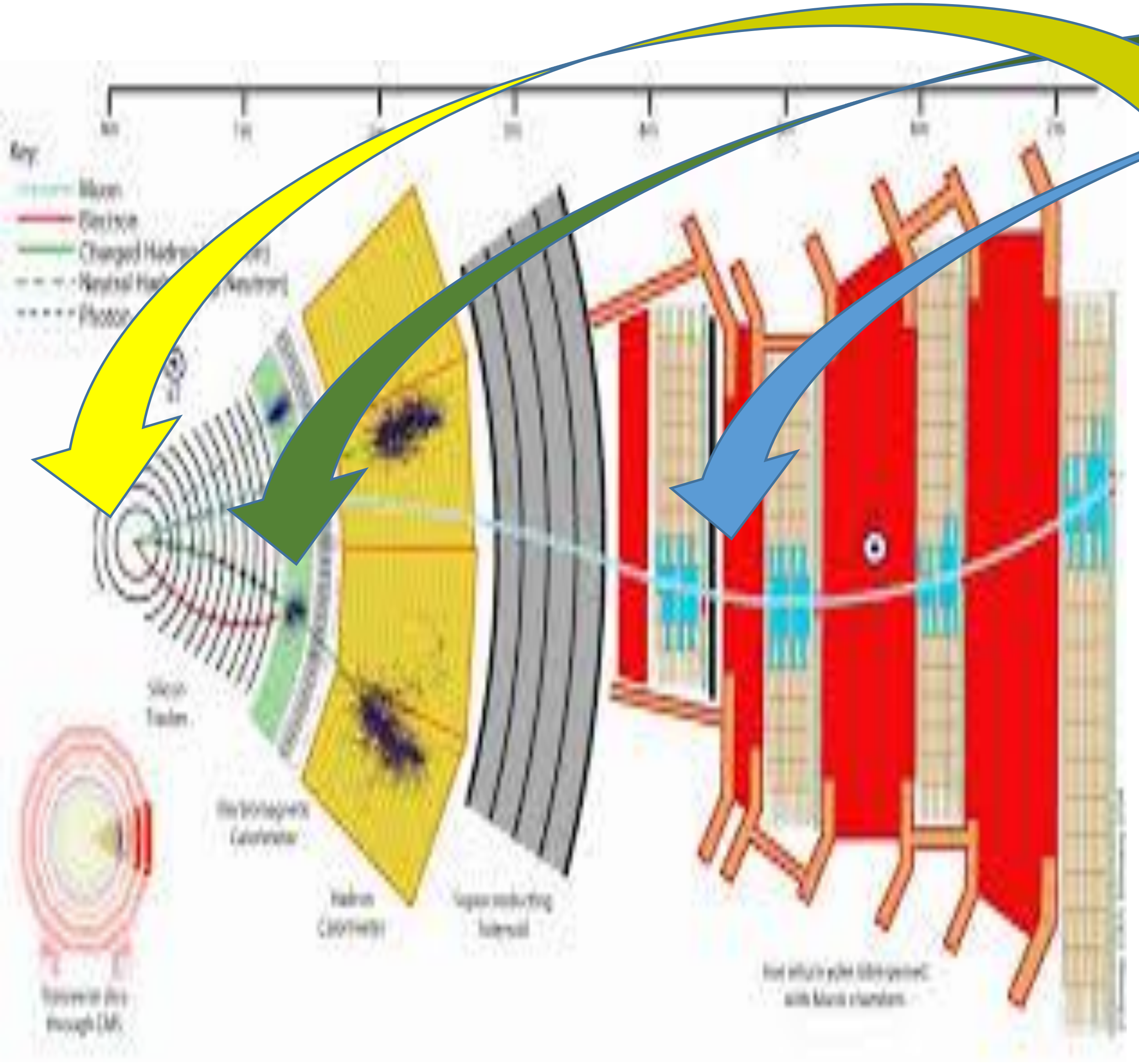
The beam crossing interval is 25 ns which corresponds to a crossing frequency of 40 MHz.

The design output rate limit of the L1 Trigger is 100 kHz, which translates to a calculated maximal output rate of 30 kHz, assuming a safety factor of 3.

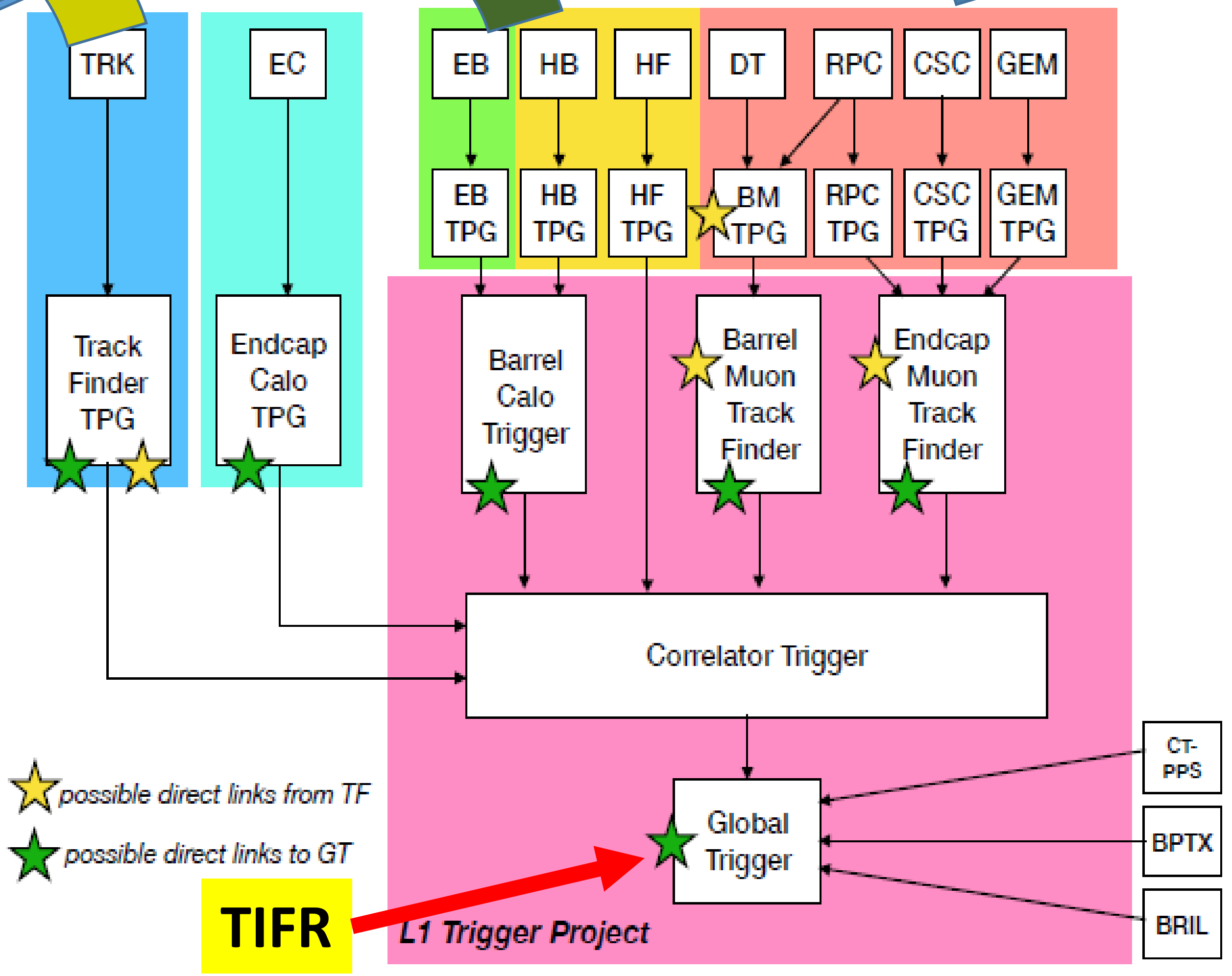
The allowed L1 Trigger latency, between a given bunch crossing and the distribution of the trigger decision to the detector front-end electronics, is 3.2 μs.

The processing must therefore be pipelined in order to enable a quasi-deadtime-free operation.

CMS Detector and the first trigger stage – Level 1



CMS detector



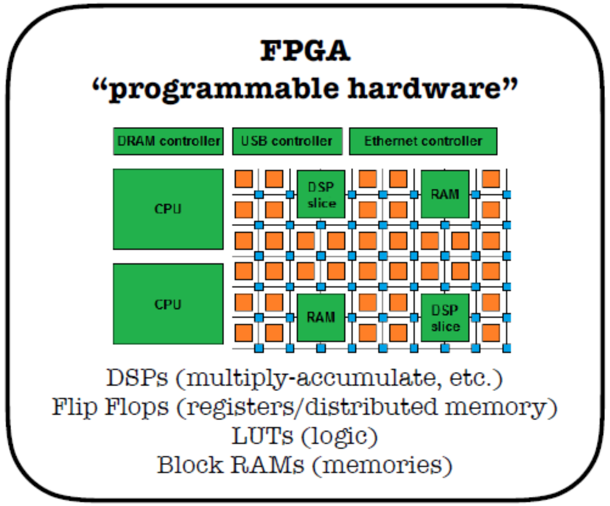
Schematic diagram – Phase-II Level-1 trigger

Level-1 Trigger upgrade – Phase II

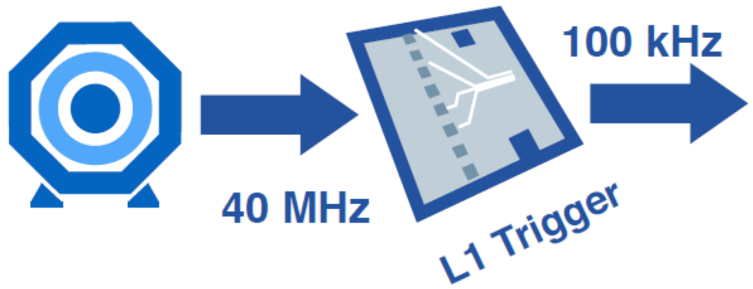
The instantaneous luminosity will increase by a factor of 5 (10^{34} cm^{-2} s^{-1} -> $5*10^{34}$ cm^{-2} s^{-1}).
There may not be enough rejection power in the current muon and calorimeter triggers.
Upgrade of the Level-1 Trigger system is needed to cope with the higher occupancies and data rates.

Several approaches will have to be combined to reach this goal:

- More complex operations at an early level, such as residual energy subtraction in the calorimeter trigger;
- More sophisticated trigger algorithms such as complex correlations between different types of trigger data, calculation of invariant masses or transverse masses of pairs of trigger objects;
- Use of information from additional parts of the CMS detector, in particular from the silicon strip tracker.



CMS
Detector

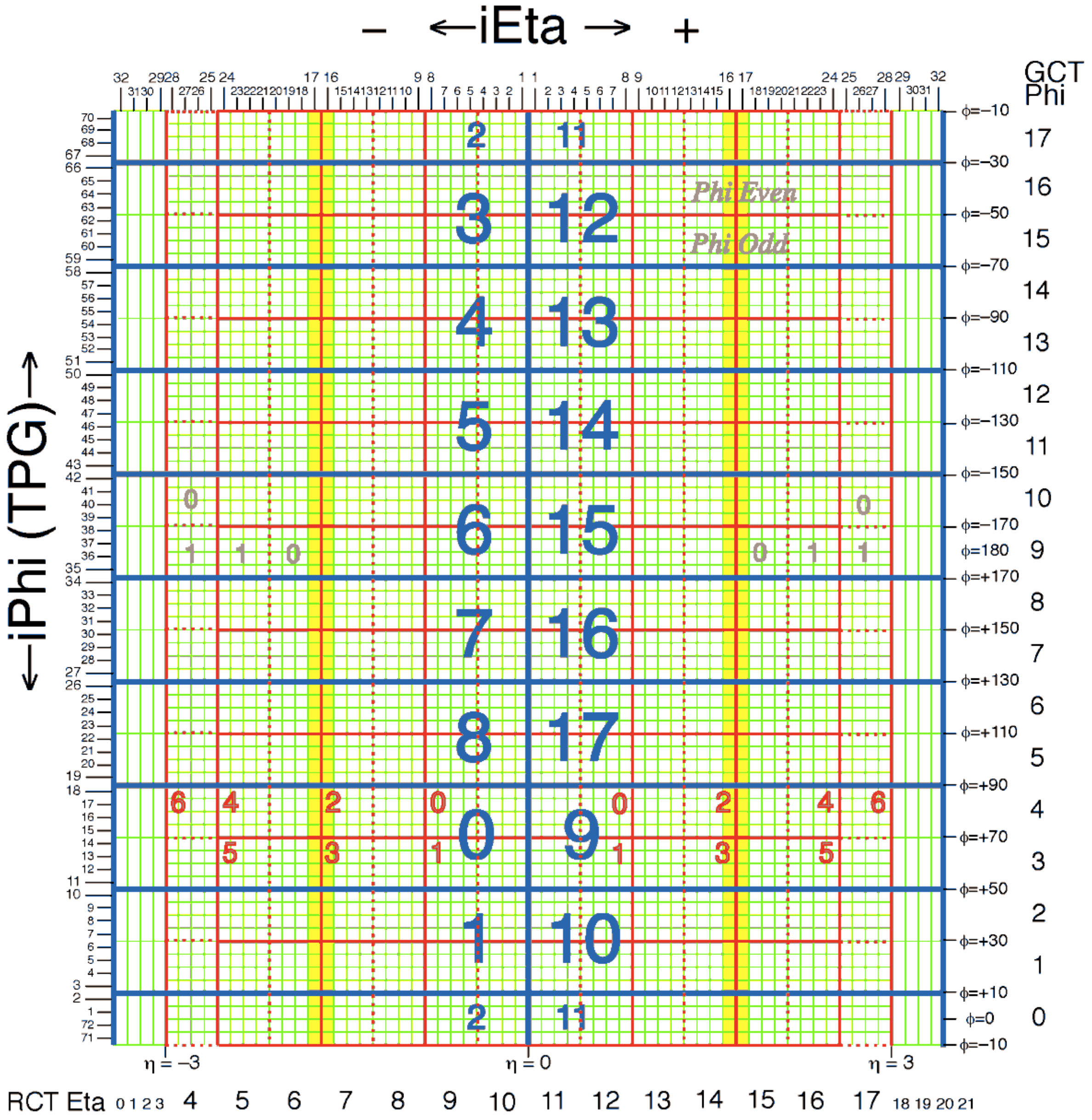


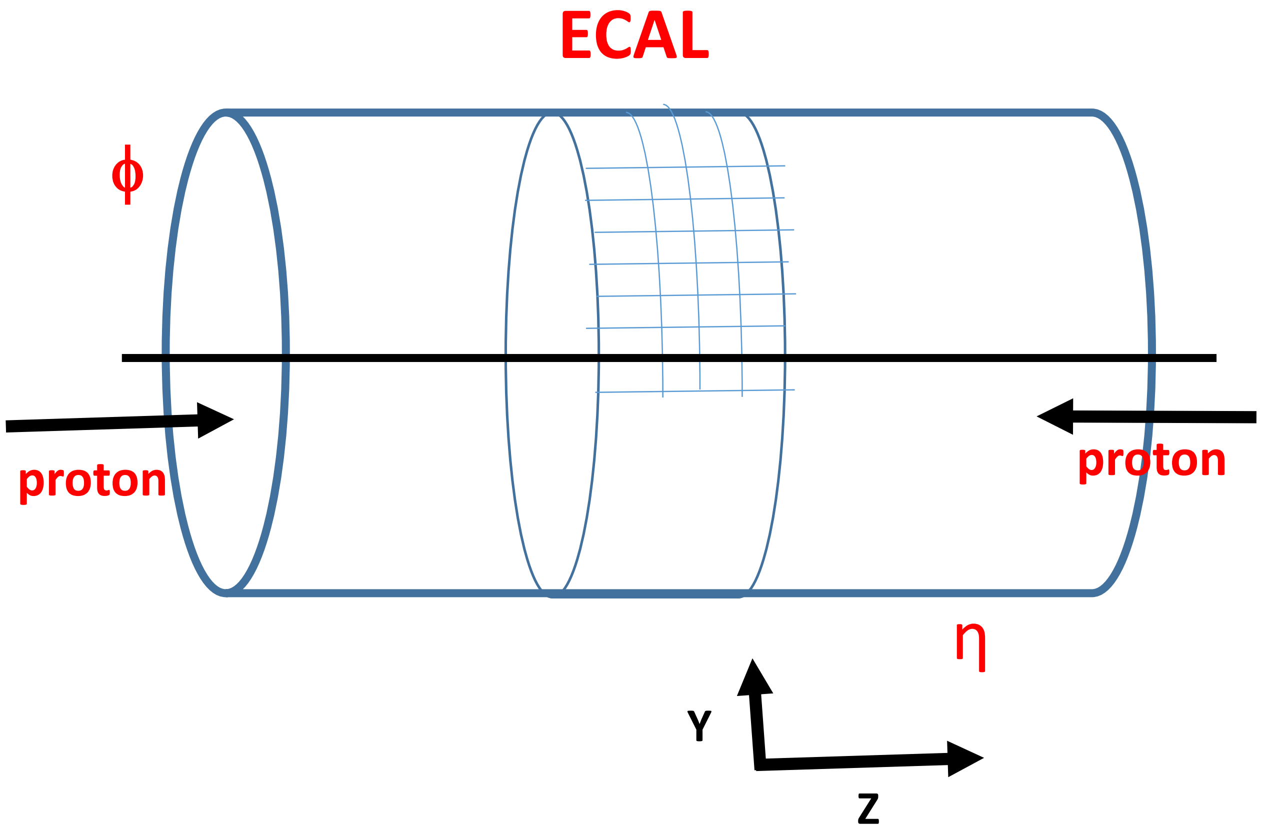
- Trigger development targets high end, powerful FPGAs
- They are becoming more accessible due to the High Level Systhesis (HLS) open CL

Introduction to High Level Synthesis (HLS)

- HLS is an automated design process that interprets algorithm specification at a high abstraction level and creates digital hardware/RTL code that implements that behavior.
- HLS significantly accelerates design time while keeping full control over the choice of optimal architecture exploration, proper level of parallelism and implementation constraints.
- Vivado HLS includes a complete design environment with abundant possibilities in the form of pragma directives to fine-tune hardware generation process from High Level Language (HLL) to Hardware Description Languages (HDL)
- HLL input languages:
  - ANSI-C (GCC 4.6)
  - C++ (G++ 4.6) → **straightforward integration in CMSSW**
  - **Etc…**
- Vivado HLS C/C++ libraries contain functions and constructs that are optimized for implementation in an FPGA.

Electromagnetic Calorimeter Layout





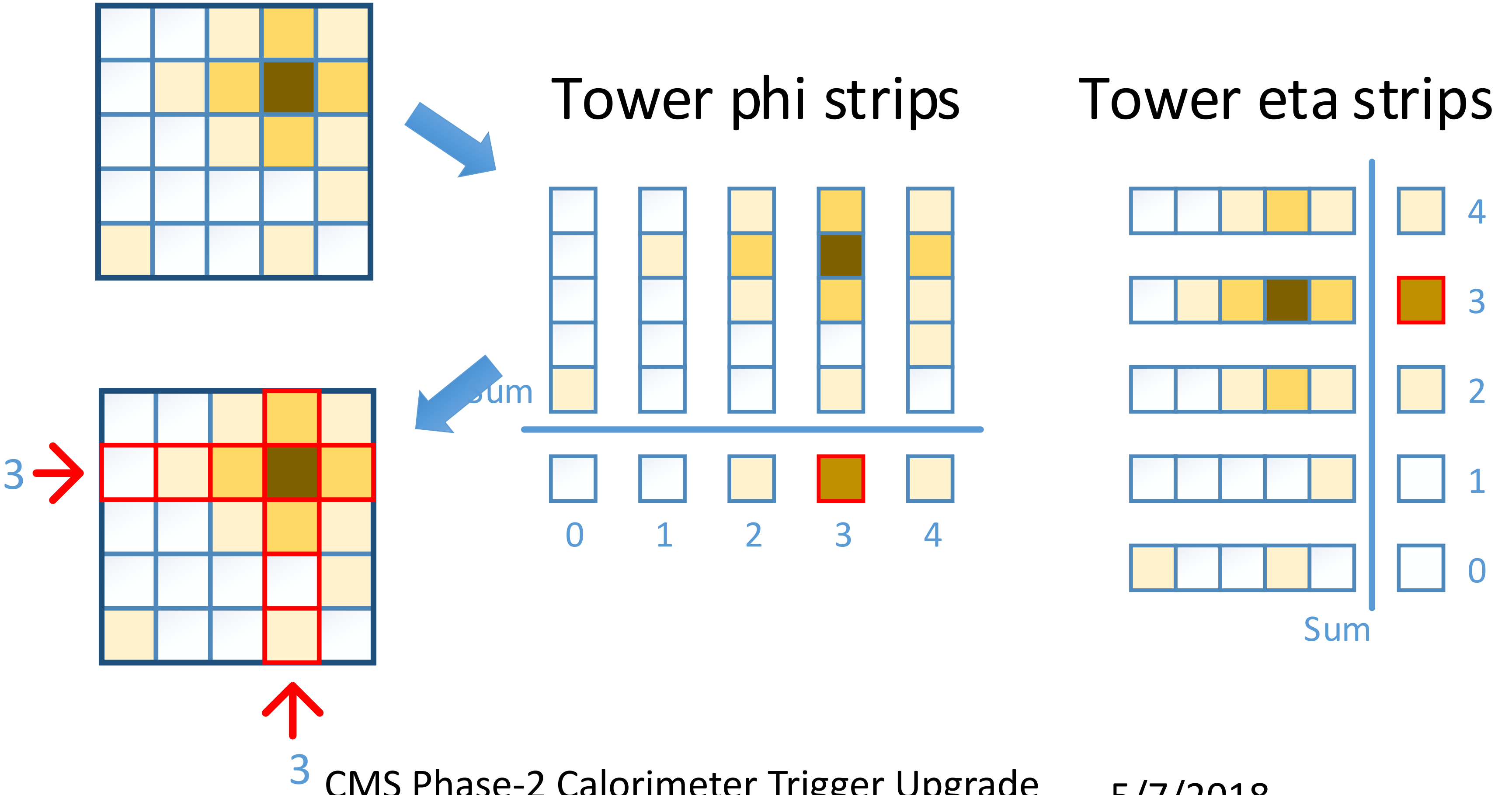
Clustering in the Electromagnetic Calorimeter

Different Algorithms for Phase 2 CaloLayer1 card and synthesized using HLS Vivado

- Input used : ($17\eta*4\phi$) towers = (17*4*5*5) crystals
- Target Device used for latency and performance study
  - Xilinx Virtex-7 690T (used in Phase I CTP7)
  - Xilinx Virtex UltraScale+ VU9P (For Phase 2, just for the illustration purpose, the exact part is still under consideration)

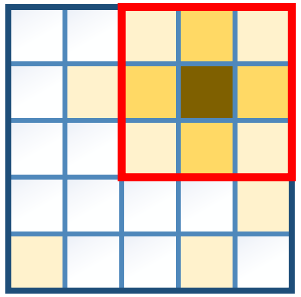
Clustering in the Electromagnetic Calorimeter

- **Step 1:** Find peak energy position for each tower (5x5 crystals) by using energy weighted sums:

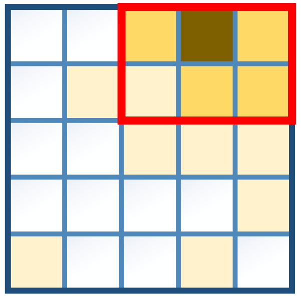


Clustering in the Electromagnetic Calorimeter

- **Step 2:** Calculate the 3x3 cluster energy around the peak energy position.



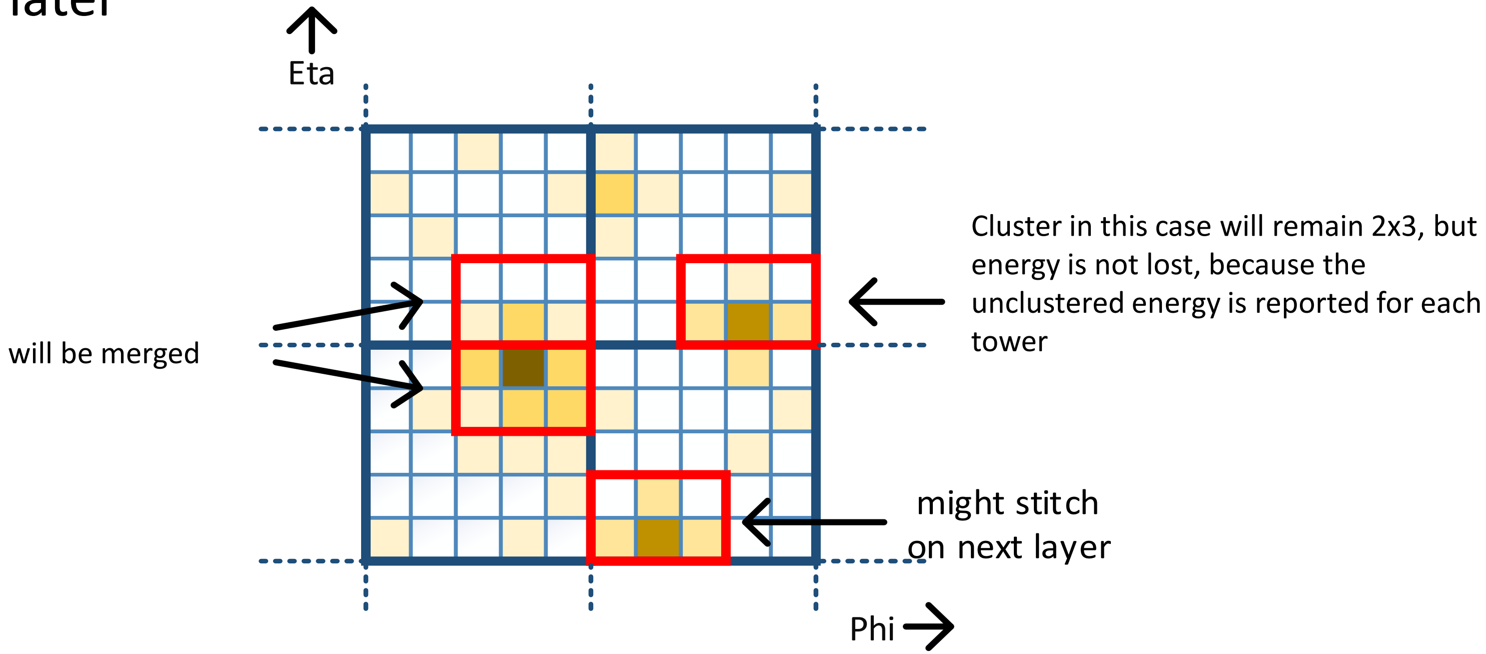
will need stitching



- Clusters which have their peaks on tower edges are stitched while considering the entire caloLayer1 card .

Clustering in the Electromagnetic Calorimeter

- **Step 3:** Clusters belonging to the same card (e.g. 17η x 4ϕ) and that have their peaks on tower edges are merged.
  - Merge the energy of smallest cluster to largest cluster
  - Assign the smallest cluster energy to be 0 GeV to avoid double counting later



will be merged

Cluster in this case will remain 2x3, but energy is not lost, because the unclustered energy is reported for each tower

might stitch on next layer

- Cluster on card boundaries are stitched in layer-2.

Egamma clustering Algorithm for CaloLayer1 card

- Step 1: Identify for every 5x5 crystal ("trigger tower”), the peak crystal position in eta-phi
  - Use energy-weighted position algorithm
- Step 2: Calculate the 3x3 cluster energy around the peak crystal.
- For each tower in a calo-layer1 card (17η*4ϕ) towers , results in :
  - Peak position in eta and phi
  - 5x5 tower sum
  - 3x3 cluster energy around the seed
- Step3: Merge the 3x3 clusters with peak energy at the tower boundaries
- *For the time being, we have only taken ECAL tower information in the algorithm (no HCAL input)*

HLS Vivado Performance

Target Clock used is : 240 MHz (4.16 ns)

Target Device : Xilinx Virtex-7 690T

Latency of 53 clock cycles and 22% (FF) and 35% (LUT) usage of resources

Absolute latency : 53(4.16/25) = 8.8BX*

```
================================================================
== Performance Estimates
================================================================
+ Timing (ns):
    * Summary:
    +--------+-------+----------+------------+
    |  Clock | Target| Estimated| Uncertainty|
    +--------+-------+----------+------------+
    |ap_clk  |   4.16|      3.64|        0.52|
    +--------+-------+----------+------------+


+ Latency (clock cycles):
    * Summary:
    +-----+-----+-----+-----+----------+
    |  Latency  |  Interval | Pipeline |
    | min | max | min | max |   Type   |
    +-----+-----+-----+-----+----------+
    |   53|   53|    6|    6| function |
    +-----+-----+-----+-----+----------+
```

```
================================================================
== Utilization Estimates
================================================================
* Summary:
+-----------------+---------+-------+--------+--------+
|       Name      | BRAM_18K| DSP48E|   FF   |   LUT  |
+-----------------+---------+-------+--------+--------+
|DSP              |        -|      -|       -|       -|
|Expression       |        -|      -|       0|   21934|
|FIFO             |        -|      -|       -|       -|
|Instance         |        -|      -|  142188|   97904|
|Memory           |        -|      -|       -|       -|
|Multiplexer      |        -|      -|       -|   26229|
|Register         |        -|      -|   53300|    9840|
+-----------------+---------+-------+--------+--------+
|Total            |        0|      0|  195488|  155907|
+-----------------+---------+-------+--------+--------+
|Available        |     2940|   3600|  866400|  433200|
+-----------------+---------+-------+--------+--------+
|Utilization (%)  |        0|      0|      22|      35|
+-----------------+---------+-------+--------+--------+
```

HLS Vivado Performance

Target Clock used is : 320 MHz (3.12 ns)

Target Device : Xilinx Virtex UltraScale+ VU9P

Latency of 72 clock cycles and 8% (FF) and 13% (LUT) usage of resources

Absolute latency : 72(43.12/25) = 8.9BX*

```
================================================================
== Performance Estimates
================================================================
+ Timing (ns):
    * Summary:
    +--------+-------+----------+------------+
    |  Clock | Target| Estimated| Uncertainty|
    +--------+-------+----------+------------+
    |ap_clk  |   3.12|      3.02|        0.39|
    +--------+-------+----------+------------+


+ Latency (clock cycles):
    * Summary:
    +-----+-----+-----+-----+----------+
    |  Latency  |  Interval | Pipeline |
    | min | max | min | max |   Type   |
    +-----+-----+-----+-----+----------+
    |   72|   72|    8|    8| function |
    +-----+-----+-----+-----+----------+
```

```
================================================================
== Utilization Estimates
================================================================
* Summary:
+-----------------+---------+-------+--------+--------+
|       Name      | BRAM_18K| DSP48E|   FF   |   LUT  |
+-----------------+---------+-------+--------+--------+
|DSP              |        -|      -|       -|       -|
|Expression       |        -|      -|       0|   29153|
|FIFO             |        -|      -|       -|       -|
|Instance         |        -|      -|  134985|   96976|
|Memory           |        -|      -|       -|       -|
|Multiplexer      |        -|      -|       -|   28647|
|Register         |        -|      -|   56513|   10095|
+-----------------+---------+-------+--------+--------+
|Total            |        0|      0|  191498|  164871|
+-----------------+---------+-------+--------+--------+
|Available        |     4320|   6840| 2364480| 1182240|
+-----------------+---------+-------+--------+--------+
|Utilization (%)  |        0|      0|       8|      13|
+-----------------+---------+-------+--------+--------+
```

More Examples:

- Estimate the missing energy in the event
- Identify tau leptons
- Link energy cluster with tracks for better particle id

All these methods will be used in order to design a better trigger at Level 1 to discard uninteresting events

Current status

- At TIFR we have setup simple algorithms for pattern recognition on our existing FPGAs (Kintex).
- These algorithms were tried on the Wisconsin system using advanced FPGAs (xilinx virtex-7 690T).
- We bought in 2017 a Kintex 7 FPGA kit with ZYNQ processor, learnt how to work with it, then tested the algorithms on it.
- Our studies in collaboration with the Wisconsin group on the use of HLS firmware for implementing trigger algorithms on FPGAs have been included in the interim document on Phase-II trigger upgrade which has been released.
- The Upgrade Trigger TDR is the next step which is due in 2020.
- New FPGA (Ultrascale+, ZYNQ) has been received recently. We will now do a comparative study of performance of different FPGA systems.

Offline algorithms:

We have been studying efficiency, response and resolution of AK8 jets with the aim of building an algorithm to trigger on events with boosted top quarks using jet substructure. This study is being compared with the AK4 jet performance. A poster is on display on this topic.

BACKUP

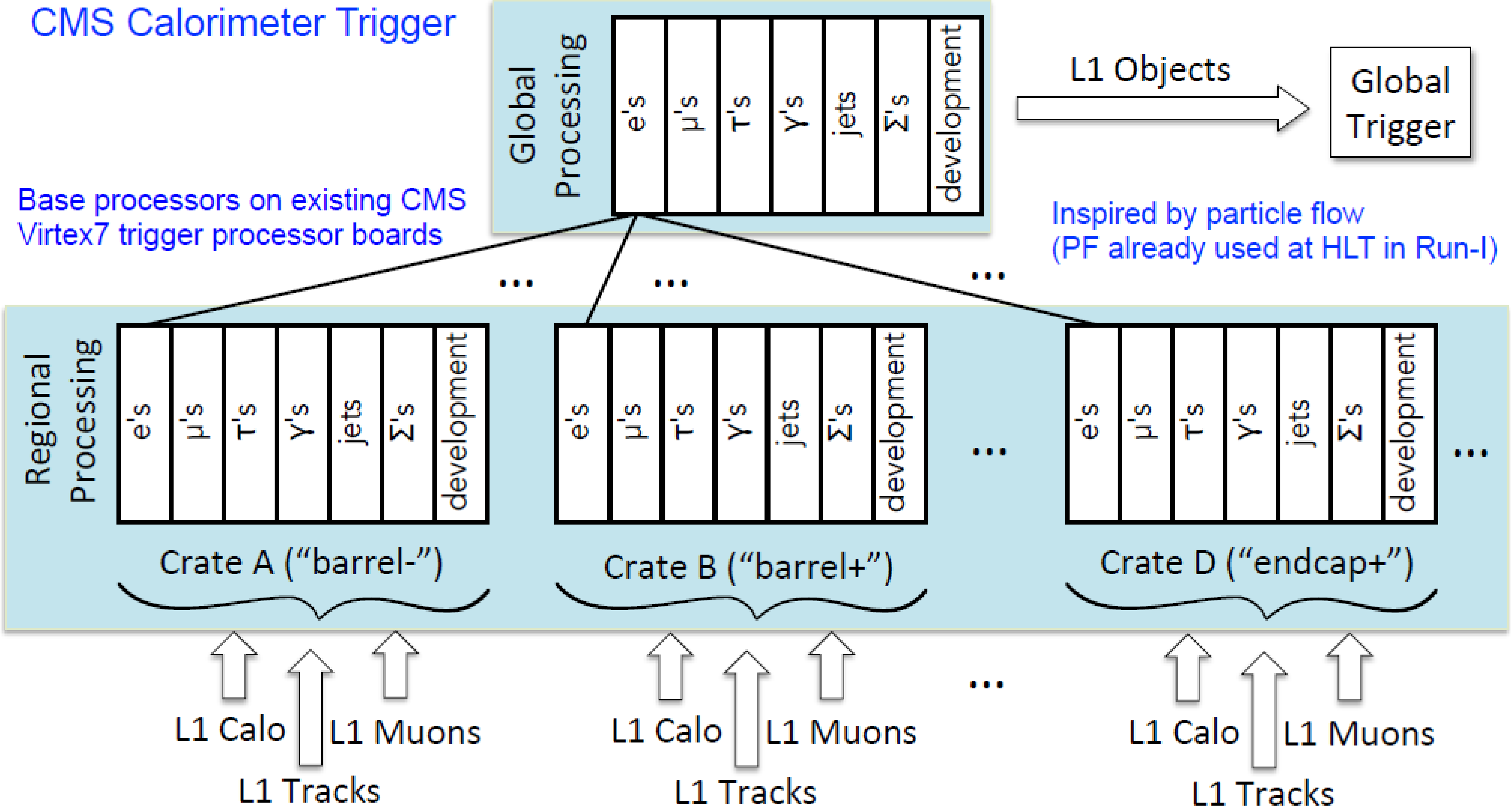
Model for L1 Correlator Trigger Hardware

Based on successful Phase 1 Architecture for CMS Calorimeter Trigger

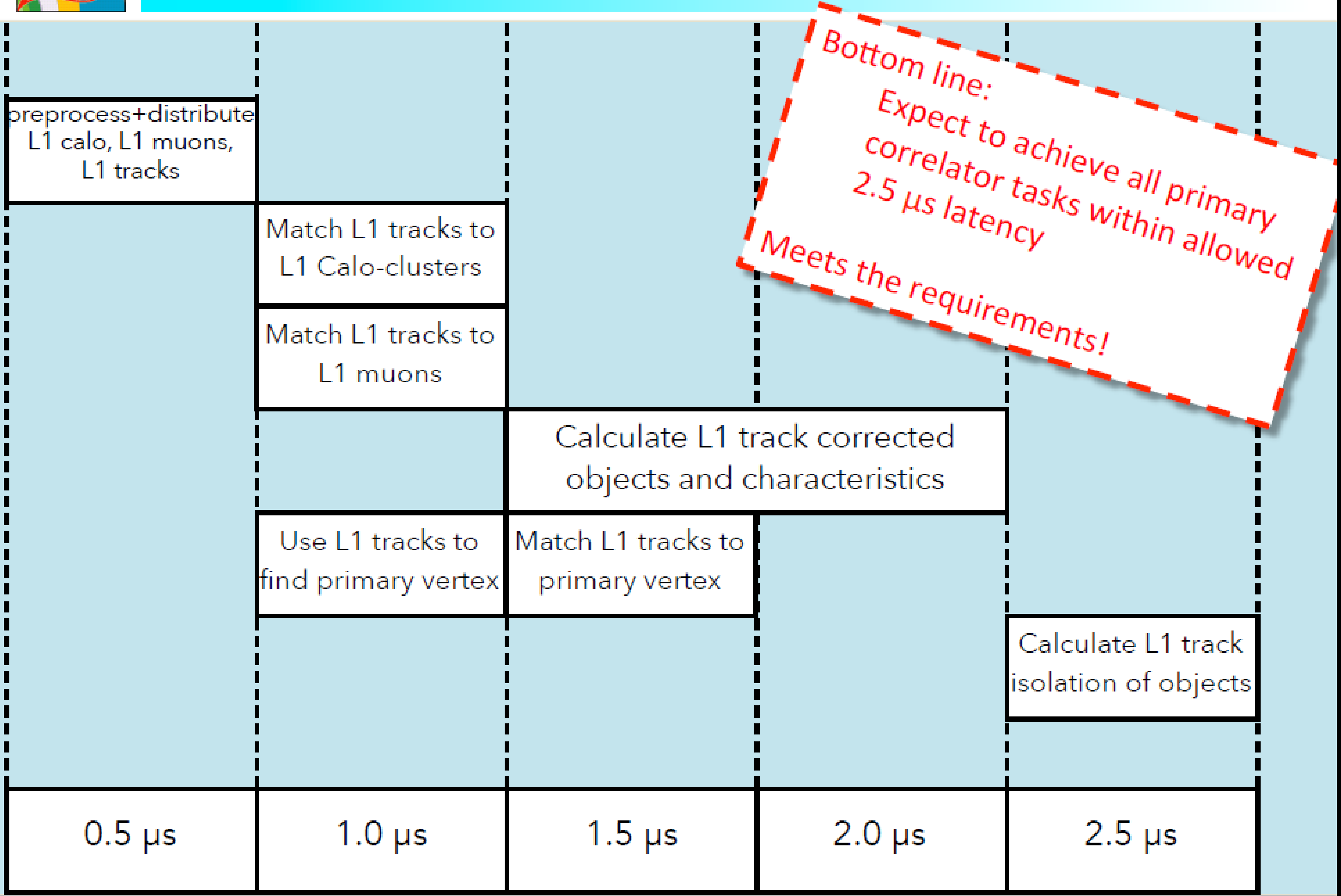
Calculate global sums (MET, SET, etc), sort objects, remove duplicates/ghosts, etc

Base processors on existing CMS Virtex7 trigger processor boards

Inspired by particle flow (PF already used at HLT in Run-I)



Correlator Trigger Workflow



preprocess+distribute L1 calo, L1 muons, L1 tracks

Match L1 tracks to L1 Calo-clusters

Match L1 tracks to L1 muons

Calculate L1 track corrected objects and characteristics

Use L1 tracks to find primary vertex

Match L1 tracks to primary vertex

Calculate L1 track isolation of objects

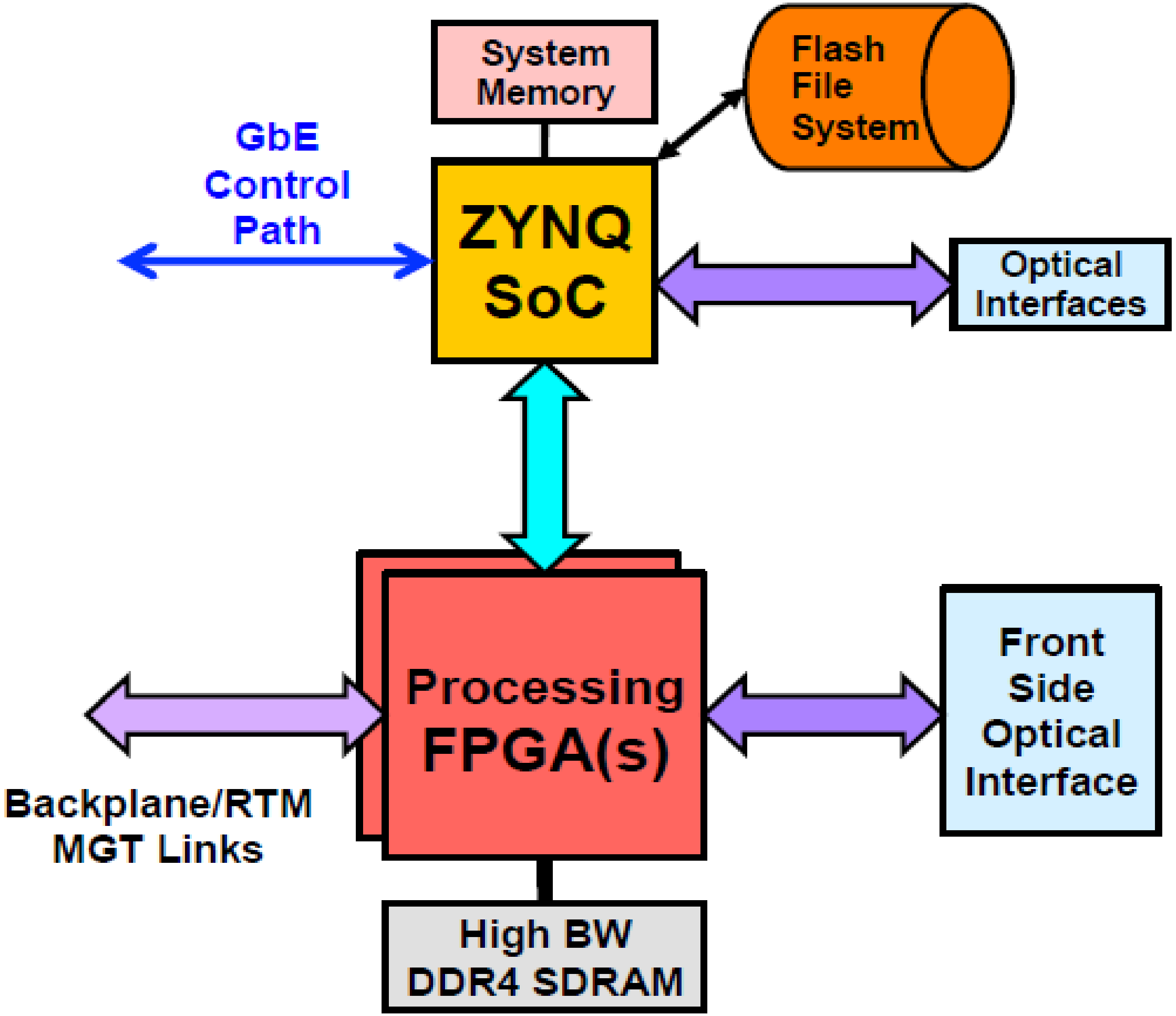
0.5 μs	1.0 μs	1.5 μs	2.0 μs	2.5 μs

Bottom line:

Expect to achieve all primary correlator tasks within allowed 2.5 μs latency

Meets the requirements!

APD Architecture Example



- Next-gen FPGA and ZYNQ SoC devices
- General upgrade to embedded Linux platform over CTP7
- Direct optical interfaces for the ZYNQ PL section
- DDR4 SDRAM on main FPGA for higher density and bandwidth
- Optical module mix for compatibility with current and next-gen optical links

- Currently, one or two system architectures for the generic trigger processor boards are being designed and evaluated by the collaboration.
- Possible arrangement of boards and links together with processing functions to be performed by each board are also being worked at.
- These functionalities will be realised by expansion mezzanines and supplemental FPGA processor boards.
- We will procure high FPGA/SoC development/evaluation kits and will initially build the functionalities mentioned above.
- Apart from helping us upgrade our technology skills, this process/method also allows us decouple risk of building high cost hardware straight away.
- The kits will help both firmware and algorithm development work.

Xilinx EK-Z7-ZC706-G development kit