Stage-2 Tau Algorithm: firmware-oriented pseudocode

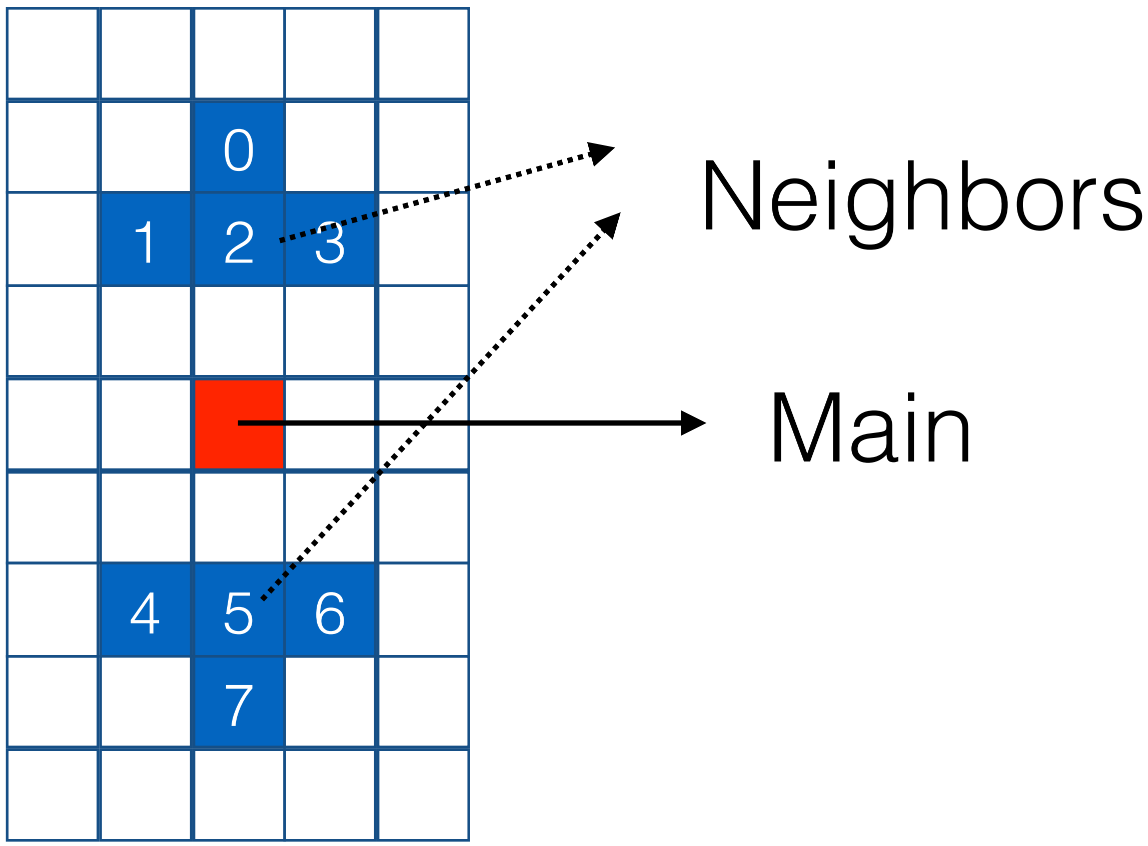
Luca Cadamuro

Laboratoire Leprince-Ringuet (École polytechnique)

Algorithm main steps

- Merging strategy → L1 tau candidate creation
  - 1: Neighbor search
  - 2: Merging
- 3: Calibration
- 4: Isolation
- 5: Shape veto

Introduction



- Squares indicate **clusters** (i.e., size can be larger than 1 TT)
  - the square indicates the cluster position, i.e. the cluster seed TT
- We work on formed clusters
- Neighbors clusters are searched only in the highlighted pattern

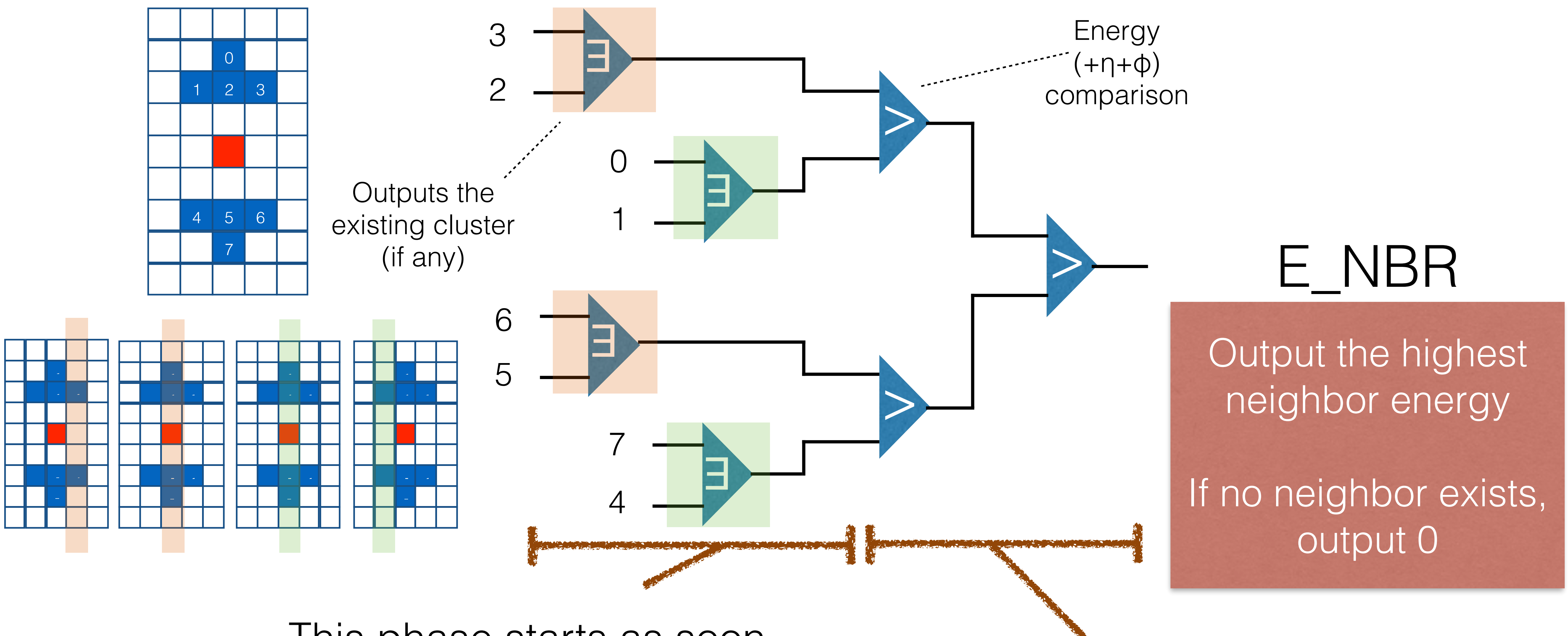
FLAGS:

- E_NBR [9]

(highest neighbor energy coded on 9 bits)

1 - Neighbor search

- Can exploit the limited number of existing neighbor patterns:



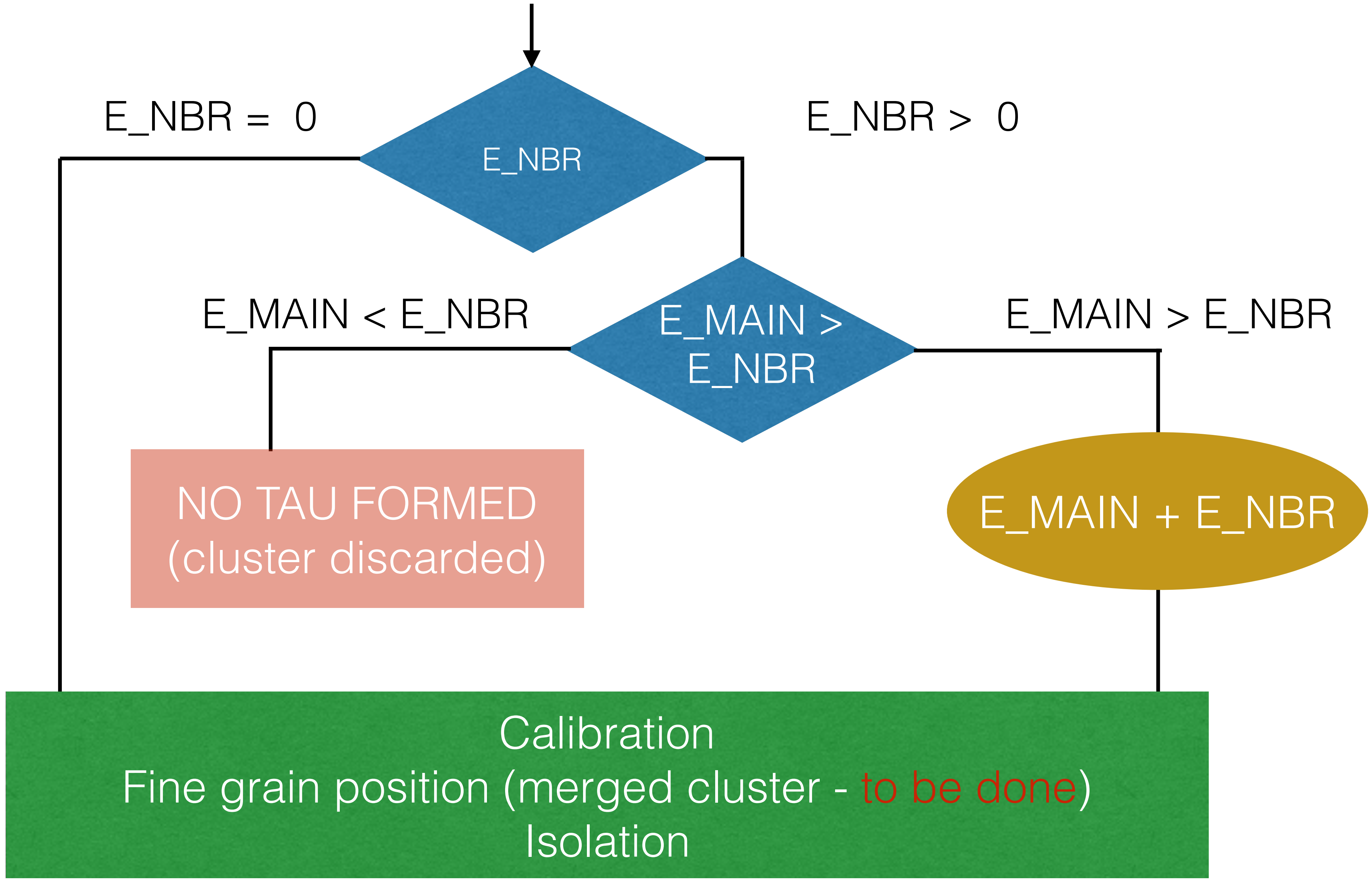
This phase starts as soon as clusters are formed, but need to wait for two clock cycle (cluster compared are shifted by 1 iη unit)

Need to wait for the energy of the clusters to be computed

2 – Merging

- Diagram to summarize the entire process

E_MAIN = E_NBR never happens because if energy is equal than eta than phi are compared (like EG algo)



3- Energy calibration

Calibration is computed as a linear function of the hadronic and EM energy deposited.

$$E_{calib} = (aE_{EM} + bE_{had} + p) \cdot d(i\eta) \qquad \text{(if } E_{EM} > 0\text{)}$$

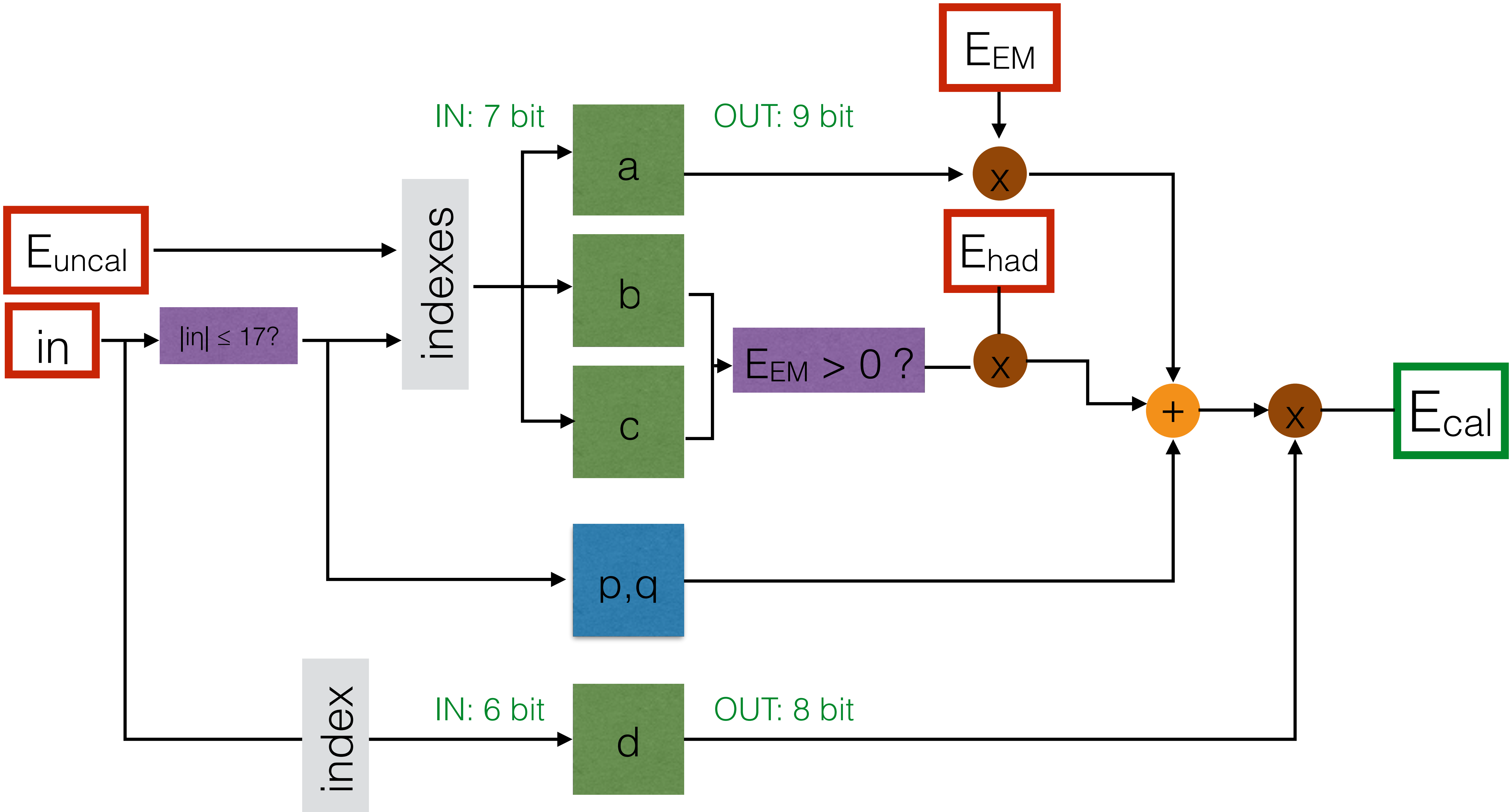
$$E_{calib} = (cE_{had} + q) \cdot d(i\eta) \qquad \text{(if } E_{EM} = 0\text{)}$$

- **a, b, c** depend on $E_{uncal} = E_{EM} + E_{had}$ and on the barrel/endcap cluster position
- **p, q** depend on the barrel/endcap cluster position
- **d** depends on the position of the cluster in iη units

Coefficient values are stored in a LUT.

In the case of merging, input E_{uncal}, E_{EM}, E_{had} are all computed as the sum of the energies of the two merged clusters.

Energy calibration – data flow



Need to check the allowed bit numbers

4 - Isolation

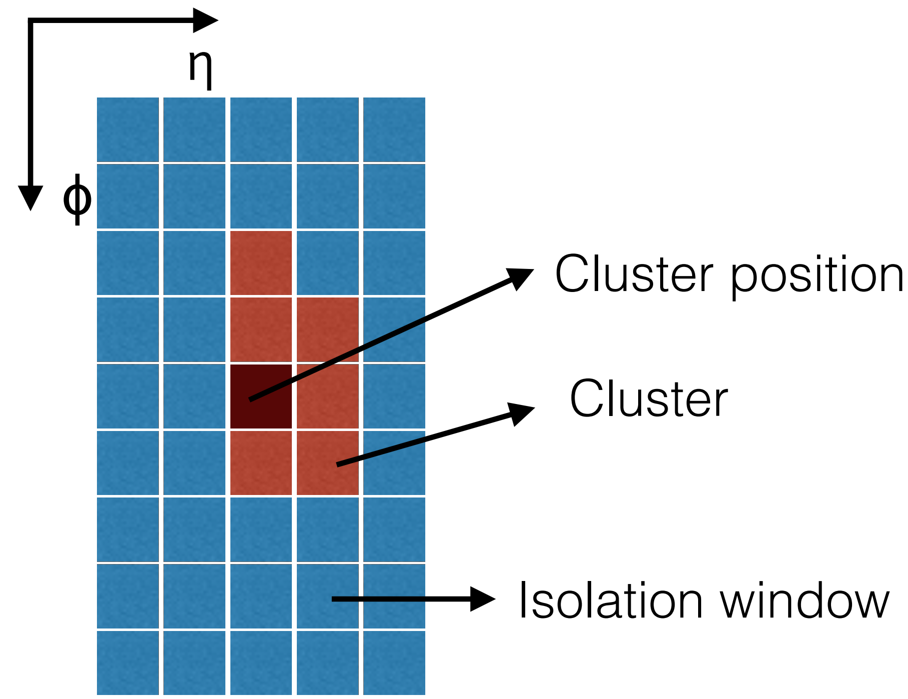
- Isolation is computed as:

$$E_{5x9} - E_{uncal} \leq IsoThr(E_{uncal}, N_{TT}, \eta)$$

  The threshold depends on the cluster energy and η as well as from the pile-up indicator N_{TT}.

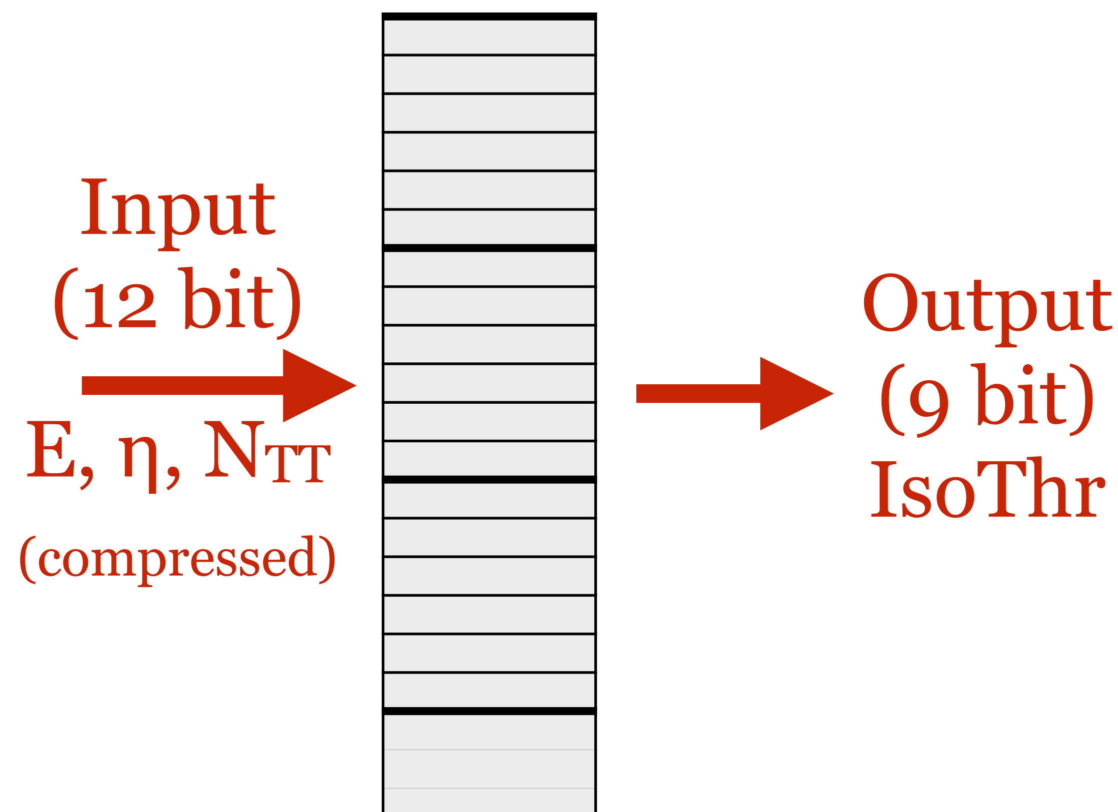
  Coded in a LUT. It could be also possible to input just (E, η) in two LUTs to get A, B coefficients to multiply externally by N_{TT} (IsoThr is a linear function of N_{TT}: IsoThr = A + B N_{TT})

  N_{TT} = num. of towers with E > 0 in the region iη ∊ [-4, +4]



- The isolation window has a size 5x9 in (η,ϕ) and is **always** centered around the cluster position (computed as the initial cluster seed position)

- Compression has already been developed and tested



5 – Shape veto

- Cluster shape is coded in the same way as EG algorithm
- For merged clusters, only the main cluster shape is used at the moment
- Vetoed shapes encoded in a LUT
  - a fixed set of shapes is used at the moment
  - maybe can be optimized as a function of eta (+ other variables?)
- Shape veto is applied at the very end of the tau algorithm (after isolation)
  - we can gain some resources by putting it before isolation computation?