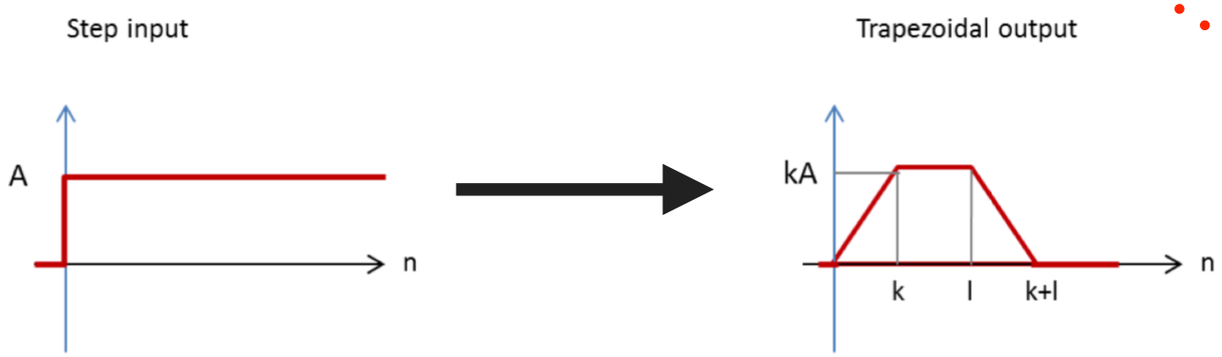
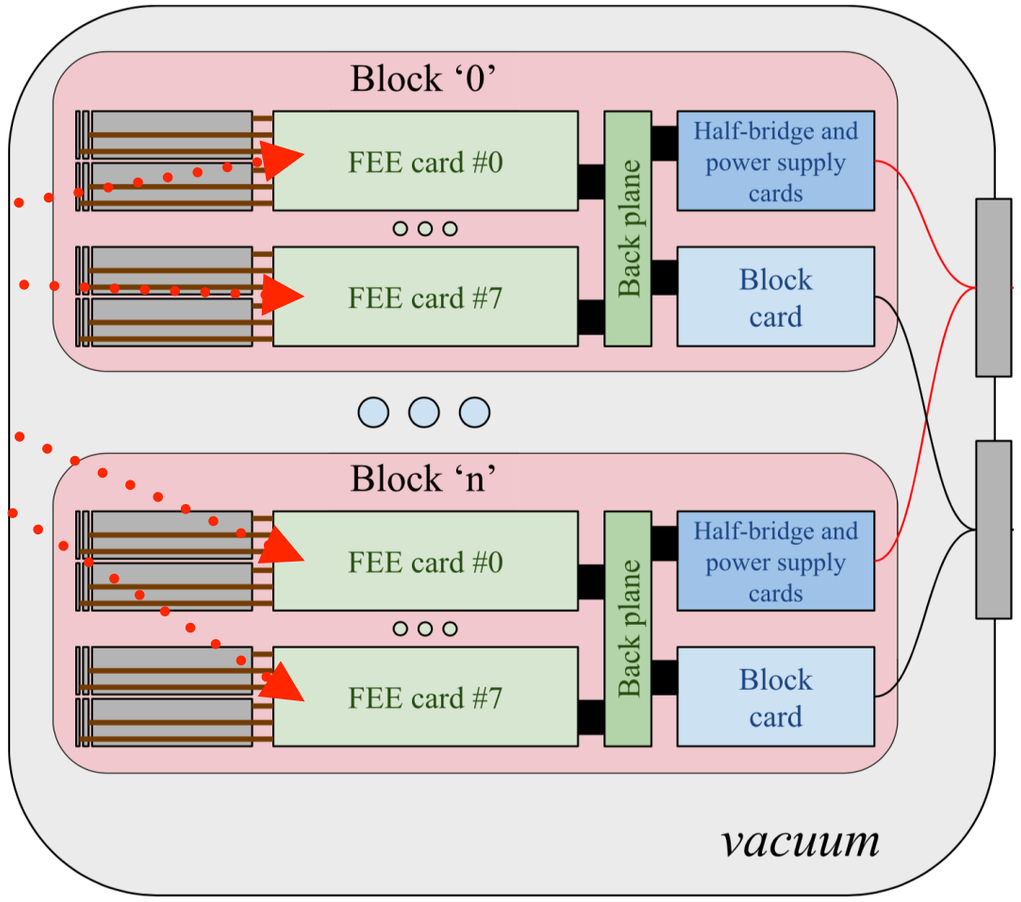
Trigger logic and Data flow

Trigger logic

Trapezoidal shaping filters on Q1, Q2 and Q3 signals



To generate local triggers



Slow control registers → **adjust** → Filter parameters (rising edge and flat top)

(200 ns rising edge / 200 ns flat top)

High/low threshold **for each channel**

- For each front-end, able to choose the trigger timeout, trigger source and the kind of trigger
- Local triggers reach the BC through 16 dedicated lines on the backplane

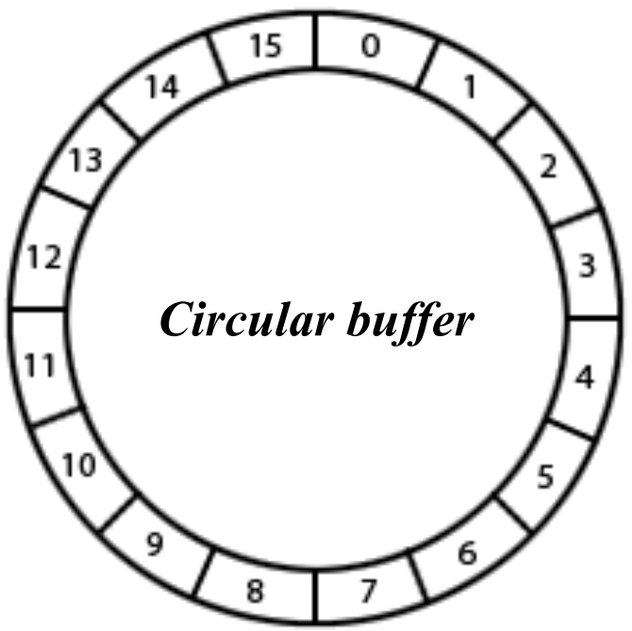
Trigger logic

- On each block, the BC counts the local triggers and send the total to the RB
- The RB applies programmable rules
- The RB will integrate the multiplicities coming only from the blocks specified by the rule, and it will produce a "rule trigger" only if overcome the multiplicity thresholds
- The RB checks if there are any alerts: FPGA data buffers are almost full, "GTT flag" from any block or there is an external veto from the "veto in" LEMO connector;
- If there is at least one alert, then a veto flag is issued. In these cases, except when there is only an external veto, the flag is also sent to the "veto out" LEMO connector.
- "trigger out" LEMO connector when there is a global trigger without the veto flag
- It decides a validation signal which is produced when "veto flag" and "global trigger" or "external trigger" from the "trigger in" LEMO connector

Data flow

FEE ⟶ BC ⟶ RB ⟶ Machine

- Four trapezoidal shaping filters on the FPGAs to calculate the energy
  - One on each Q1 and Q2 :
    - Rise time of 2 μs and flat time of 1 μs
  - Two on Q3 :
    - CsI has a “fast” and “slow” filters to identify the light ions that stop in the last stage of the telescope
    - Rise time / flat stop : 2 μs / 10 μs for slow filter and 2 μs / 500 ns for fast filter
- All signals are continuously stored in circular buffers with $N \leq 1024$
- Circular buffer
  - Data will be stored on any empty element (starting location is not important)
  - Use FIFO logic
  - When it is full, it starts overwriting the oldest data



Data flow

- With a validation on FPGAs, raw signals move to FIFO memories
- At the same time, check the threshold and marked for acquisition
- FIFO memory is used to store the whole local event
- If the telescope is marked for acquisition, 6 + 4 wave forms and event number are transferred
- Two 400 MHz serial busses between FEE and BC → 800 Mbit/s
- The busses are connected to FPGA “A”, the data from FPGA “B” must pass through FPGA “A”
- If any of the FIFO memories is about to fill up, the Global Trigger Throttle is raised and the RB is vetoed

Data flow

From FEE to BC

- When a BC FPGA receive a validation, it starts to read the FIFO memories
- Check the event number coming from each FEE
  - Discard if the number if less than expected
  - If greater than expected, skip and keep for the next event
- In this way the BC builds a coherent partial event and stores it in a FIFO buffer waiting to transfer it to the RB

From BC to RB

- All the same procedure as from FEE to BC
- RB builds a full coherent event and stores it in a large FIFO buffer
- RB adds a trigger information to allow the calculation of the dead time
- Finally, RB will send an event to first computer, and then second, and so on
- The maximum throughput is about 800 Mbit/s