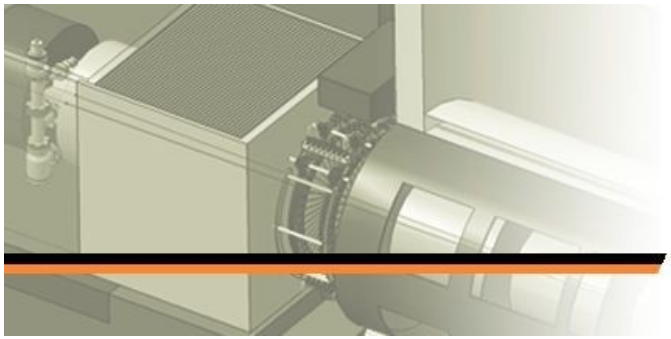
TB 2021 empty events issue – FLAME timestamp counter damage

Jakub Moroń, M. Firlej, T. Fiutowski, M. Idzik, A. Skoczeń, K. Świentek

AGH University of Science and Technology



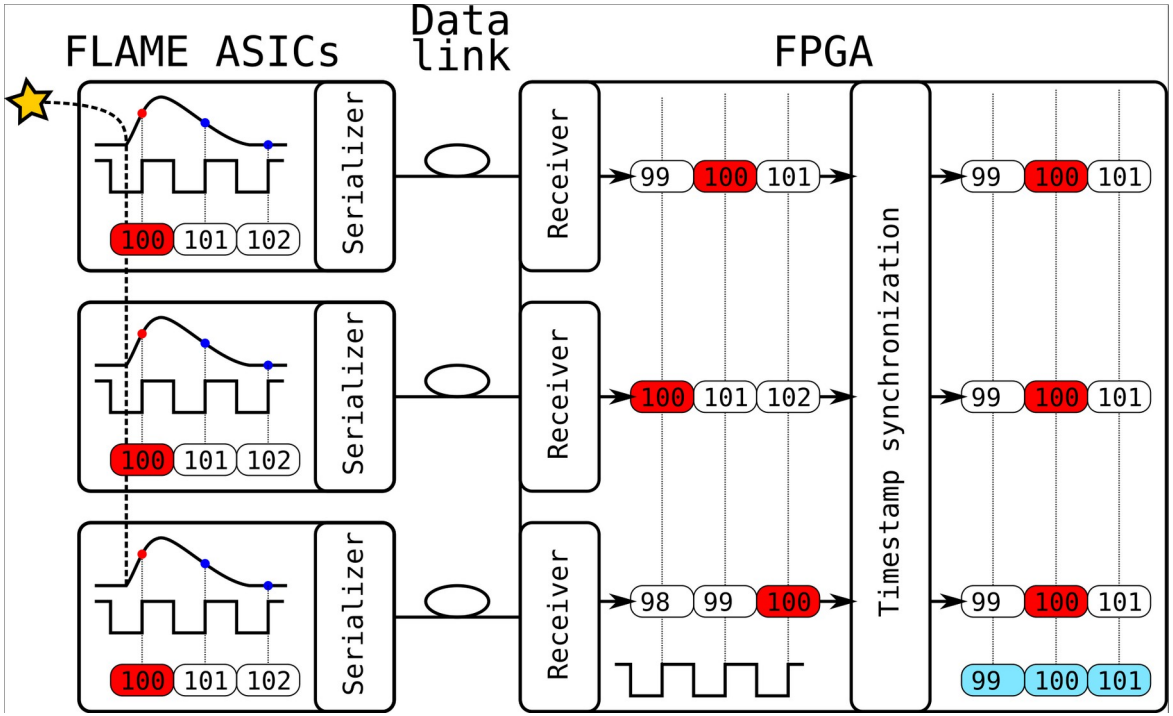
FCAL Collaboration
High precision design

AGH

FLAME timestamp

FLAME to FPGA time synchronization mechanism works as follows:

- Small 8b timestamp counter embedded in each FLAME ASIC
- ASIC counters are fully synchronized with each other and with the FPGA
- Event is read by all ASICs at the same time and labelled with the same timestamp
- Each ASIC has its own serializer → data link → FPGA receiver chain, and each chain has a different latency. So, data appears in FPGA misaligned.
- FPGA re-aligns the data from FLAMEs to the internal FPGA timestamp counter (48b) relying on the timestamp labels assigned by the ASICs



FLAME timestamp

FLAME to FPGA time synchronization mechanism was fully verified and checked multiple times. It was working flawlessly, until ESD event on the begin of the TB2021.

- Debug function added to DAQ → records the raw FLAME timestamp sequence (before alignment) and sends it in place of the regular data.
- Issue found → 5th bit of the FLAME timestamp is always zero. Therefore the timestamp sequence is as follows:

```
Expected: 0, 1, …, 30, 31, 32, 33, …, 62, 63, 64, 65, …, 94, 95, 96, 97, …, 126, 127, 128, 129, …
Recorded: 0, 1, …, 30, 31,  0,  1, …, 30, 31, 64, 65, …, 94, 95, 64, 65, …,  94,  95, 128, 129, …
```

Data properly synchronized

Data “from future”, send back in time due to the wrong timestamp

- Timestamp synchronization module in the FPGA was **not** designed to work with such a pattern (and never verified against such situation) and it was working against us, and with a bug...

FLAME wrong timestamp synchronization

```
Expected: 0, 1, …, 30, 31, 32, 33, …, 62, 63, 64, 65, …, 94, 95, 96, 97, …, 126, 127, 128, 129, …
Recorded: 0, 1, …, 30, 31,  0,  1, …, 30, 31, 64, 65, …, 94, 95, 64, 65, …,  94,  95, 128, 129, …
```

- Timestamp synchronization module synchronizes to the oldest data (with the lowest timestamp)

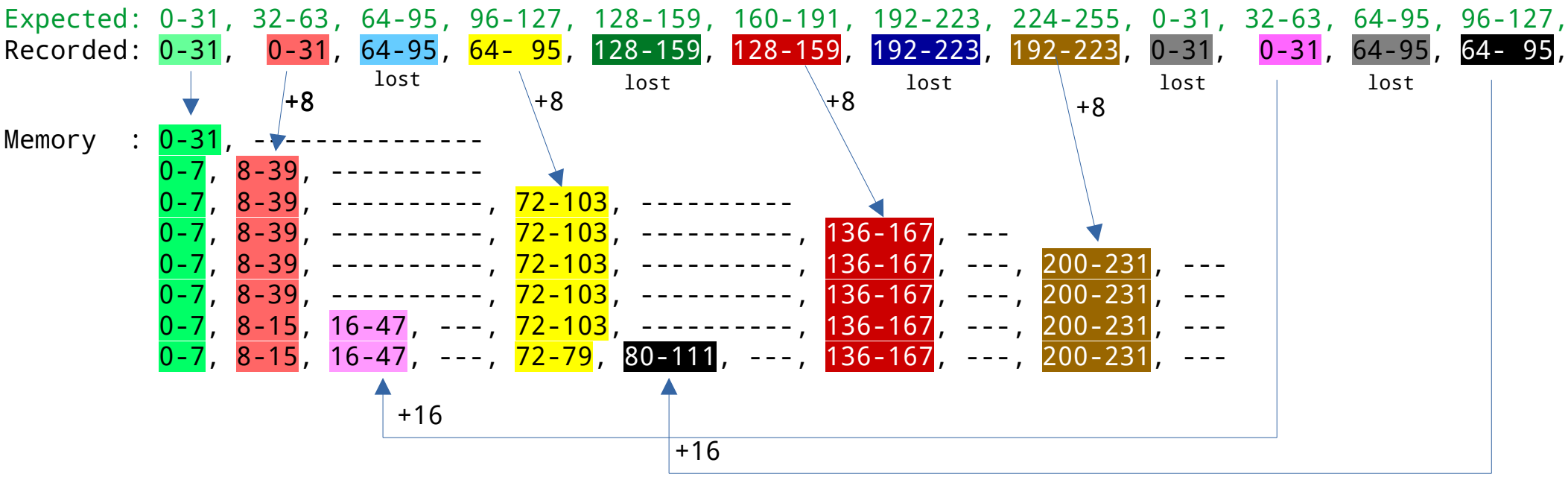
1) The green block is received earlier (in real time) than the red one
2) The red one has first sample with lower timestamp value (0) than the last one from the green (31)
3) Therefore the red block is recognized as “older” than the green one and module synchronizes to the red one

4) When the blue block starts to be received, there is jump of +32 in the timestamp
5) Synchronization module waits therefore 32 clock cycles to regain synchronization, holding the whole blue block in the buffer
6) But after the 32 clock cycles the yellow block starts to be received, and (as explained in points 1–3) is recognized as “older” and preferred over the blue one which is lost

➔ Additionally there is a bug: each time the timestamp jumps by -32 (from green to red, from blue to yellow), the block recognized as “older” (red, yellow) is shifted by **8, 16, 24** or **32** samples (see next slide) into the future due to the FIFO overflow.
Moreover, shift of more than 8 samples is taken into account and partially compensated by the trigger circuitry:
Shift by 8 → remains 8
Shift by 16, 24 and 32 → compensated to 8… (but not 0!)

FLAME wrong timestamp synchronization

- Due to the improper FLAME timestamp counter, data is written into 32 sample chunks



- All the data written into memory is “from the future” and is shifted by **2, 2.4, 2.8 or 3.2 us** in respect to the real time.
- After four cycles (4*256 = 1024 samples) all the memory is filled up with mix of current data (but still shifted in time) and residues from previous cycles (also shifted)

Due to the partial compensation of the buggy shift done by the trigger circuitry, all samples are effectively shifted by **+2 us → 2x the DESY II revolution period of 1 us** *(https://particle-physics.desy.de/test_beams_at_desy/e252106/e252334)*

Summary

- On the very beginning of the TB2021 we have got an ESD event – timestamp counters in ASICs are significantly damaged and are not counting sequentially, but in 32 samples chunks (5th bit stuck at zero)

- FPGA synchronization module was not designed for non-continuous timestamps and it works “peculiarly” → selects the chunks shifted in time and adds its own shift, partially compensated by the trigger circuitry

- Effectively, all the data is shifted by **2 us:**

  - Since the DESY II revolution period is 1 us, if there is an additional beam particle after the one which generated the trigger, it will arrive 1, **2**, 3,… us after the initial one.

  - The 2 us delay matches perfectly the data shift, and I saw in the raw ADC data the signals located +/-1 us around trigger (see talk from the 2021.12.16 meeting) with roughly the same probability (~0.9%) as the one “centered” on the trigger.

  - This means (if my understanding is correct) that all our data originates from the additional particles not related to any trigger → **we cannot use the telescope data...**