ECL DAQ status

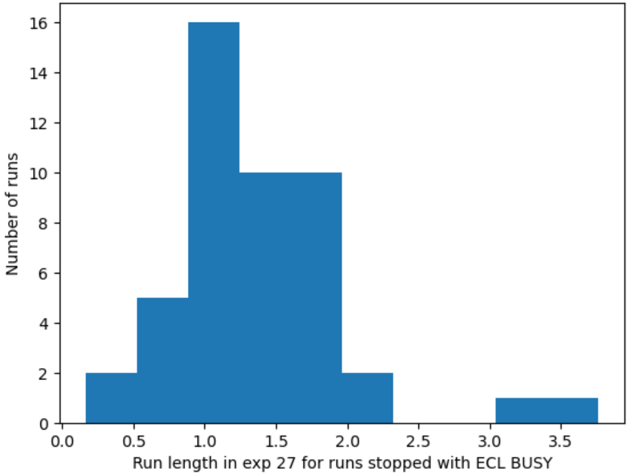
Belle II Trigger/DAQ workshop, 2022.11.29

Mikhail Remnev

ECL persistent busy in PCIe40 readout tests

* During DAQ high-rate tests with 30 kHz poisson trigger, we have been seeing frequent run stops due to ECL busy.

* On average, runs are stopped after ~1 hour.



* ECL busy signal seems to be due to a buffer overflow in PCIe40. Surprisingly, busy always comes from the PCIe40 corresponding to ECL crates B19-B36.

- This is not an issue of a particular board, different setups have been tested to confirm this.

* From testing different configurations, we can see that saving larger amounts of ECL waveform data causes higher frequency of ECL busy.

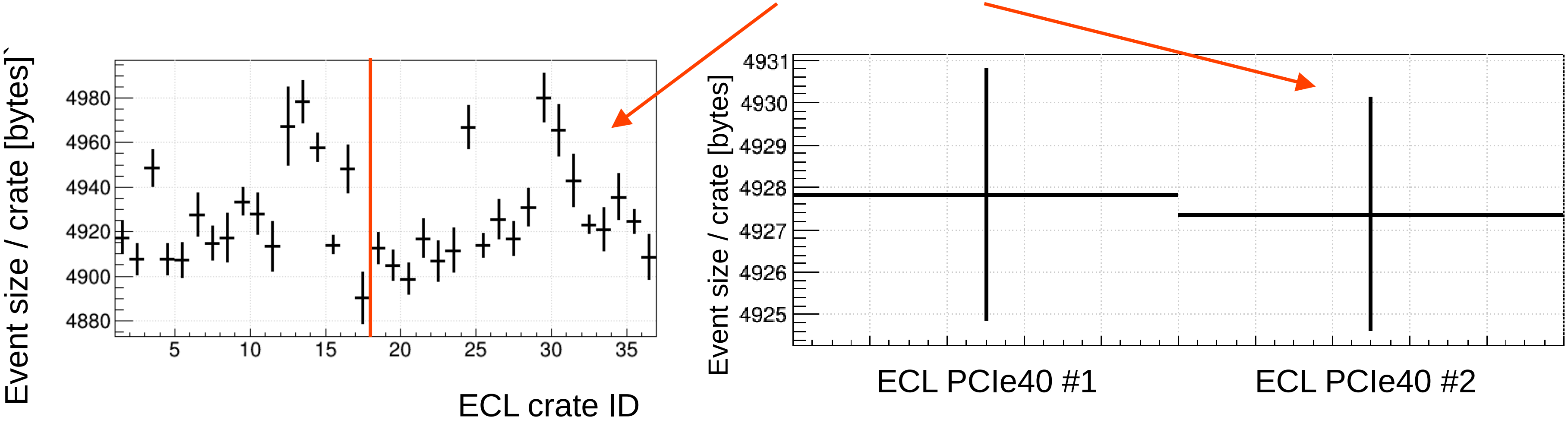
- no waveform saving → no ECL busy (3 kbytes/s/board)
- 100% waveform saving → ECL busy in ~10 minutes (80 kbytes/s/board)

ECL persistent busy in PCIe40 readout tests

* I am a bit surprised about these results. If we save 100% of waveform data, the average data size from ECL crates 1-18 is not much different from ECL crates 19-36.

* Waveform data is compressed by ECL FEE modules.
Higher electronics noise => lower compression efficiency.
Thus, noisy channels result in larger event size.

ECL busy comes from PCIe40 corresponding to these crates



ECL persistent busy in PCIe40 readout tests

* In any case, we have some measures to prevent buffer overflow:

- There is an automated measure that blocks waveform saving if there is a risk of buffer overflow in ECL FEE.

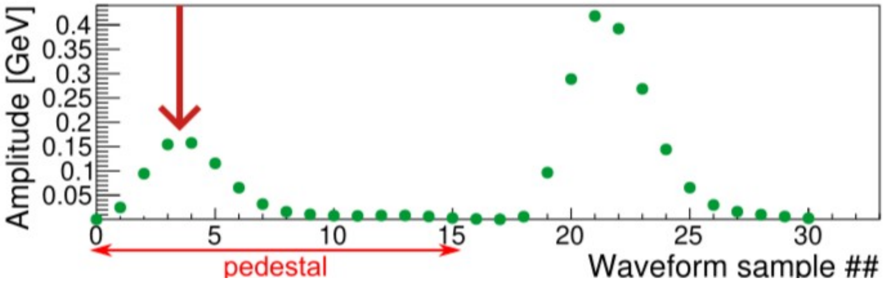
- There is a buffer in ECLCollector module, so I think we can handle delays of at least up to 15 μs when sending data to PCIe40 without losing anything.

Handling increased size of waveform data

Handling increased size of waveform data

As was reported in TB meeting on 2022.11.18, injection background causes data loss in ECL due to underestimation of the hit amplitude.

- Higher pedestal → lower amplitude → more hits below the 1 MeV threshold are discarded.



This can be fixed in several ways, best energy resolution is achieved if we save waveform data with bad pedestal for offline re-processing at the ECL unpacking stage.

However, as this leads to the increase in the number of saved waveforms, we are also considering ways to reduce ECL data size.

Handling increased size of waveform data

* One possibility is to reduce (possibly to 0) the number of waveforms with E > 50 MeV saved for pulse shape discrimination, as pulse shape discrimination algorithm can be done within ECL FEEs.
(using more sophisticated fit algorithm)

* The algorithm is being prepared and tested.
However, there is an additional issue – the algorithm will require more configuration parameters, so total FEE configuration size will be ~500 MB instead of ~300 MB.

* That is a possible issue because we are using the same configuration in ECL DQM for data validation.
- We fit some fraction of ECL waveforms within ECL DQM and compare the results with ones obtained from ECL FEE.
- This feature is vital for quickly noticing possible issues with ECL electronics. (and has helped us multiple times)

Handling increased size of waveform data

* Currently: ~300 MB per basf2 process at HLT.
Possible new value: ~500 MB per basf2 process at HLT.

* 300 MB is already not very good, I would prefer not to increase it further.

* One option is to do this part of ECL DQM procedure only at ExpressReco.
- Is it fine to have large RAM usage there?

* Other alternative is to do data processing on dedicated ECL PC by sending event data with the scheme similar to the scheme utilized by EventDisplay.

Summary

* Two main conflicting issues in ECL DAQ are

1. Large size of waveform data.
2. Handling of injection background (that might likely require to save more waveform data).

We are trying to reconcile the solutions to these issues.