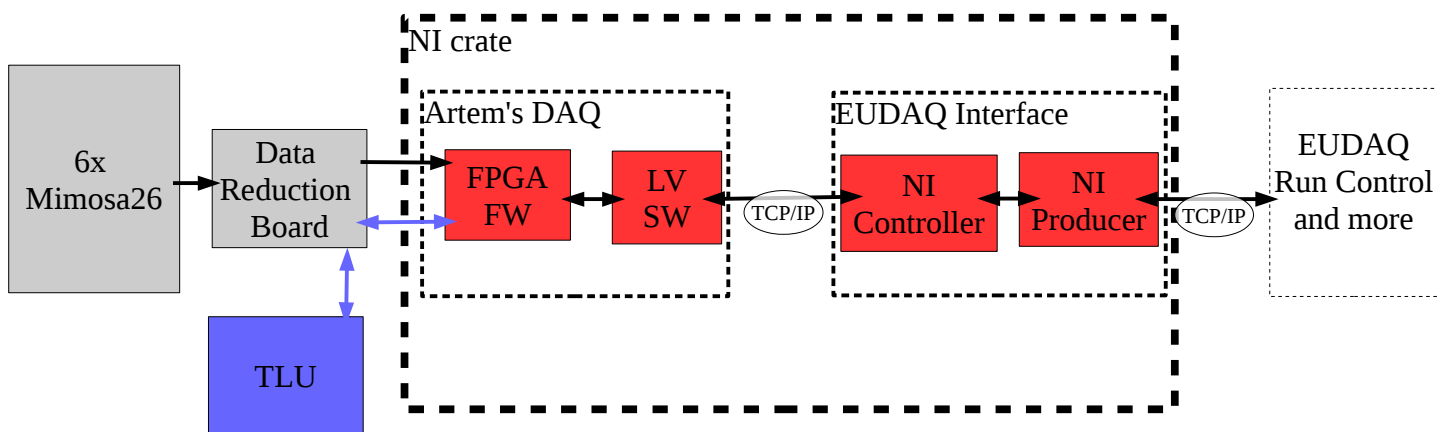


Mimosa26 (NI crate) integration in EUDET telescopes by Artem (~2011/2012) reconstructed by Hendrik Jansen and Jan Dreyling-Eschweiler, 7th Januaray 2016

Information

We didn't look into the FPGA FW and LV SW up to now. However, we learned the functionality of the EUDAQ interface (NI Controller and Producer), as well as, we interpreted the data stream which arrives at the NI Controller from the LabView SW with the help of the ConverterPlugin, which decodes the data stream.

Overview



In red: current (Artem's) integration → to be replaced

In blue: TLU communication Trigger-Data-Handshake

Conclusion (in brief)

- Between LV SW and NI Controller, one data stream consists of 6 simultaneous frames of each sensor/plane (sensor 0 to sensor 5 = 6 planes)
 - One data stream consists of
 - Header (4 bytes): “Header0” (2 bytes) and “Header1” (2 bytes)
 - Only for the first sensor (!): TriggerID (2 bytes) and PivotPixel (2 bytes)
 - something like Framecounter (4 bytes) or Framecounter (2 bytes) and 2 bytes empty
 - something like "Data length0" and "Data length1"
 - Hits: if data length not zero, StatusLines and States
 - “Trailer 0” and “Trailer1”
- for sensor 0 and the same for sensor 1 to 5
- NI Producer takes two consecutive of these data streams to build one EUDAQ event (= two times AddBlock.)

Details and references

- FPGA firmware (FPGA FW):
Binary: anemone.lvbitx Code: ???
- LabView software (LV SW):
Code: anemone_v1.3.lv

- EUDAQ NI Controller

Code: <https://github.com/eudaq/eudaq/blob/v1.5-dev/producers/ni/src/NiController.cc>

- handles the TCP/IP communication with LV SW:
 - sending:
 - *NiController::ConfigClientSocket_Send* (l. 118):
for starting and stopping the sensor read out
 - receiving:
 - *NiController::DataTransportClientSocket_ReadLength* (l. 250):
data length to know how long is the consecutive data stream
 - *NiController::DataTransportClientSocket_ReadData* (l. 284):
data stream of 6 simultaneous frames for the 6 planes
- see print outs of these methods in the attached txt-files:
- `example_nicontroller_dataflow_thr12_1kHz_autotrigger.txt`
 - **`one-event_nicontroller_dataflow_thr12_1kHz_autotrigger.txt`**
 - `only-trigger-framecounter_nicontroller_dataflow_thr12_1kHz_autotrigger.txt`

- EUDAQ NI Producer

Code: <https://github.com/eudaq/eudaq/blob/v1.5-dev/producers/ni/src/NiProducer.cxx>

- handles the EUDAQ event building:
 - In *MainLoop()* (l. 24), two consecutive of these data streams are used for one EUDAQ event.

- EUDAQ ConverterPlugin for encoding the raw data stream

Code: <https://github.com/eudaq/eudaq/blob/v1.5-dev/main/lib/plugins/NiConverterPlugin.cc>