

Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie

AGH UNIVERSITY OF SCIENCE AND TECHNOLOGY

Jakub Moron *on behalf of FCAL/ECAL-P* Calorimeter report

LUXE Computing and DAQ meeting 2023.04.26



- Very short FCAL history
- Latest FCAL readout architecture
 FLAME ASIC
 FCAL DAQ
- ECAL-P overview
- ECAL-P readout:
 - " FLAXE ASIC (FLAME successor)
 - DAQ concept



A little bit of history...

- FCAL (Forward CALorymetry)
- Established ~2002 to develop Luminocity Calorimeter, called "LumiCal", for the International Linear Collider (ILC)



- Around 2012 FCAL joined CLICdp at CERN, adapting LumiCal to CLIC requirements
- Because both linear colliders (ILC and CLIC) have seemed increasingly unlikely over the years, FCAL switched to R&D on general purpose, compact (small Moliere radius) electromagnetic calorimeter for future experiments
- Our knowledge and experience allowed us recently to join LUXE, to build ECAL -P
- Before joining LUXE, for our R&D a universal, most flexible readout and DAQ system was needed, mainly for testbeams at DESY, but also to use cosmic muons as a signal source



- Dedicated luminosity calorimeter **LumiCal**:
 - 2 barrels on opposite sites of main detectors system
 - Each barrel 30 layers of tungsten + silicon detectors
 - Each layer 12 sensors with 4 sectors each
 - Each sector divided into 64 radial pads
 - → 3072 channels on single layer
 - \rightarrow 92 160 channels on the entire barrel







Target: a readout system which:

- Can provide highest event rate possible (testbeam time is precious),
- Have the most flexible DAQ (to be easily adapt for different triggering / beam structure), therefore fully FPGA-based with absolute minimum functionality hardwired in the readout ASIC
- Allow to read asynchronous detector signals (e.g. cosmic muons)
- Simplifies data analysis by online data pre-processing





FLAME ASIC





FLAME ASIC



- 32 mix-mode channels comprising:
 - Variable gain front-end
 - 10-bit SAR ADC, working at 20
 MSps

Data encapsulation and 8b/10b coding (according to the Xilinx MGT specification)
Multi-phase PLL based fast serializer working nominally at 5.2 Gpbs

ASIC works as waveform digitizer, continuously sampling and streaming front-end output



FLAME DAQ



- 256 channels / layer
 - B FLAME ASICs
 - 16x 5 Gbps data links
 - 80 Gbps / layer data stream
- DAQ built as FGPA cluster, with one FPGA per layer
- Zyng UltraScale+ devices used, hardware based on Trenz **TE0808 FPGA modules**
- Complete hardware for the system never fully developed, as we joined LUXE in the meantime



FLAME DAQ – testbeam hardware



TB 2019 - 2021:

- Three layers
- FPGA modules hosted by Trenz TEBF0808 development boards
- Additional PCB
 providing interface to
 the FEBs and TLU





FLAME DAQ architecture



- Constant stream from FLAME ASICs (80 Gbps) received and processed by the FPGA
- Raw ADC samples and processed data stored in FPGA memory
- On trigger (TLU or internal) variable size package (from 7 to 9869 32b words) transferred via DMA to the onboard DDR RAM, and passed (by the Linux) to the upstream UDP





For asynchronously sampled CR-RC pulse the amplitude and timing are reconstructed using deconvolution procedure:

- ADC samples filtered by 3-sample FIR filter
- Two non-zero FIR samples produced for FE pulse





12/23

AGH



DAQ scheme



stored in FPGA block RAM circular buffers. 20 Mb of RAM used (~62% of the available 32Mb) *Much more than really needed to accommodate for the TLU latency*

• Last 2048 data samples

- Package built on trigger (TLU, self or command):
 - Zero suppressed
 reconstructed data
 (only channels with signal)
 - Optionally up to 63 raw data samples

- Minimal package: $7 \times 32b = 224 b$
- Maximal package: 9869 x32b = 315'808 b
- Average size: 158 x32b = 5'056 b
- Maximal event rate: ~25 kevents / s (125 Mbps)



Backend – EUDAQ producer



- Dedicated computer for FLAME data acquisition
- Run fully controlled from the main EUDAQ control instance
- Event data from FLAME DAQ stored locally in a ROOT tree on the dedicated computer storage.
- On parallel to each event stored locally, a dummy event with a proper TLU number, but no FLAME data, is sent to the main EUDAQ data collector in order to allow the EUDAQ to complete event
- Online data quality monitoring running on a dedicated computer



ECAL-P overview

Final ECAL-P design still under development:

- Originally based on GaAs sensors from Tomsk University (128 channels each)
- For obvious reason sensors changed to CALICE Si sensors (256 channels each)
- 6 sensors per layer:
 1'536 channels per layer
- 15 or 20 layers: either
 - 23'040 channels / ECAL-P
 - 30'720 channels / ECAL-P





- ECAL-P readout is going to utilize components of the existing system, adapted for LUXE timing scenario:
- FLAME ASIC → FLAXE ASIC:
 - The same front-end and 10b SAR ADC in each channel
 ADC sampling rate can vary, from ~10 up to ~40 MSps, with 20 MSps nominal rate
 - 32 channels / ASIC
 - DAQ embedded in ASIC (see next slides)
 - So called "power pulsing" developed: putting majority of the ASIC into the sleep mode between bunch crossings to reduce power consumption
- Zynq UltraScale+ based DAQ, but with single device for the whole ECAL-P
 - Signal pre-processing and reconstruction performed online, but sequentially between bunch crossings, 128 (or 256) channels at once



FLAXE – ECAL-P readout ASIC



- ADC continuously samples the front-end output
- Data from ADC collected in RAM-based circular buffer, 64 samples depth
- Acquisition stopped after bunch crossing by FPGA (ACQ_CTRL signal)
- Variable amount of samples (1 64) read from buffer via I²C-like interface



FLAXE acquisition modes



Experiment:

- FPGA enables ASIC some time before bunch crossing, which powers up analogue front-end
- After some programmable delay the ADC is started
- Acquisition starts after few clock cycles
- Data is collected in circular buffer until FPGA de-asserts ACQ_CTRL on trigger / BX clock
- FPGA reads data

Testbeam / debug:

- Front-end and ADC always enabled
- ACQ_CTRL directly control acquisition to the internal memory



- Each layer (1'536 channels): 48 FLAXE ASICs
- ASICs on a single plane connected in parallel to the common data bus
- After each BX, FPGA reads sequentially data from all ASICs on the plane
 - Max. amount of data per ASIC: 64 samples x 384 b = 24'576 b
 - ^o 48 ASICs / plane = 1'179'648 b/plane /BX
 - 10 BXs per second = \sim 12 Mb / plane / second
- 20 MHz serial clock is totally enough to read the whole plane between BXs even for the largest data package
- ECAL-P planes (15 or 20) read by FPGA in parallel





- Currently used Zynq UltraScale+ device is able to (keeping utilization and timing very relaxed):
 - Pre-process ~2.5G raw ADC samples per second
 - Reconstruct 10k of complete events per second for 128 channels = 1.28M channel depositions per second
- For ECAL-P:
 - 10 BX/s, 20 planes, 48 ASICs/plane, 32 channels/ASIC, 64 raw ADC samples / BX gives:
 - ~ <20M raw ADC samples per second for pre-processing (1% of capabilities)</p>
 - ~310k depositions (*at 100% occupancy*) (25% of capabilities)
- Data collected from the whole ECAL-P (30'720 channels) can be pipelined and processed by the DSP with 128 parallel processing channels
- Single 1Gbps UDP link to the EUDAQ is enough for ECAL-P data upstream



Since ECAL-P readout works by default with asynchronous signals, we basically do not have any timing requirements on trigger / BX clock.

However <u>having a precise BX clock (within < 1ns)</u> enables possibility to work in synchronous mode, which can, potentially, significantly boost background rejection in ECAL-P (still under investigation).

What is the expected clock / trigger distribution scheme for LUXE?





Overall scheme





- ECAL-P readout heavily based on existing FCAL readout system
- New FLAXE readout ASIC, comprising of well-tested mixed mode part from FLAME ASIC and a new back-end, sent currently for fabrication
- General concept of the new DAQ prepared, implementation will start soon
- Main question: what is / will be the expected clock / trigger distribution scheme for LUXE?
- New EUDAQ producer will be needed, we do not have an expert anymore in the collaboration...