Integration of the SAMPA in the SRS frontend

Updates on the project for the SAMPA ASIC in the SRS ecosystem

Marco Bregant

on behalf of the many people that contributed

G. G. A. de Souza, T. S. Abelha, T. B. Saramela, C. G. Penteado

IFUSP – São Paulo

and

all the colleagues from EPSUSP (São Paulo) that composed the team that developed SAMPA

RD51 MiniWeek: WG5 - Electronics for MPGD

SAMPA Design Specifications Summary

SAMPA is an ASIC developed for the readout of ALICE TPC and MCH detectors:

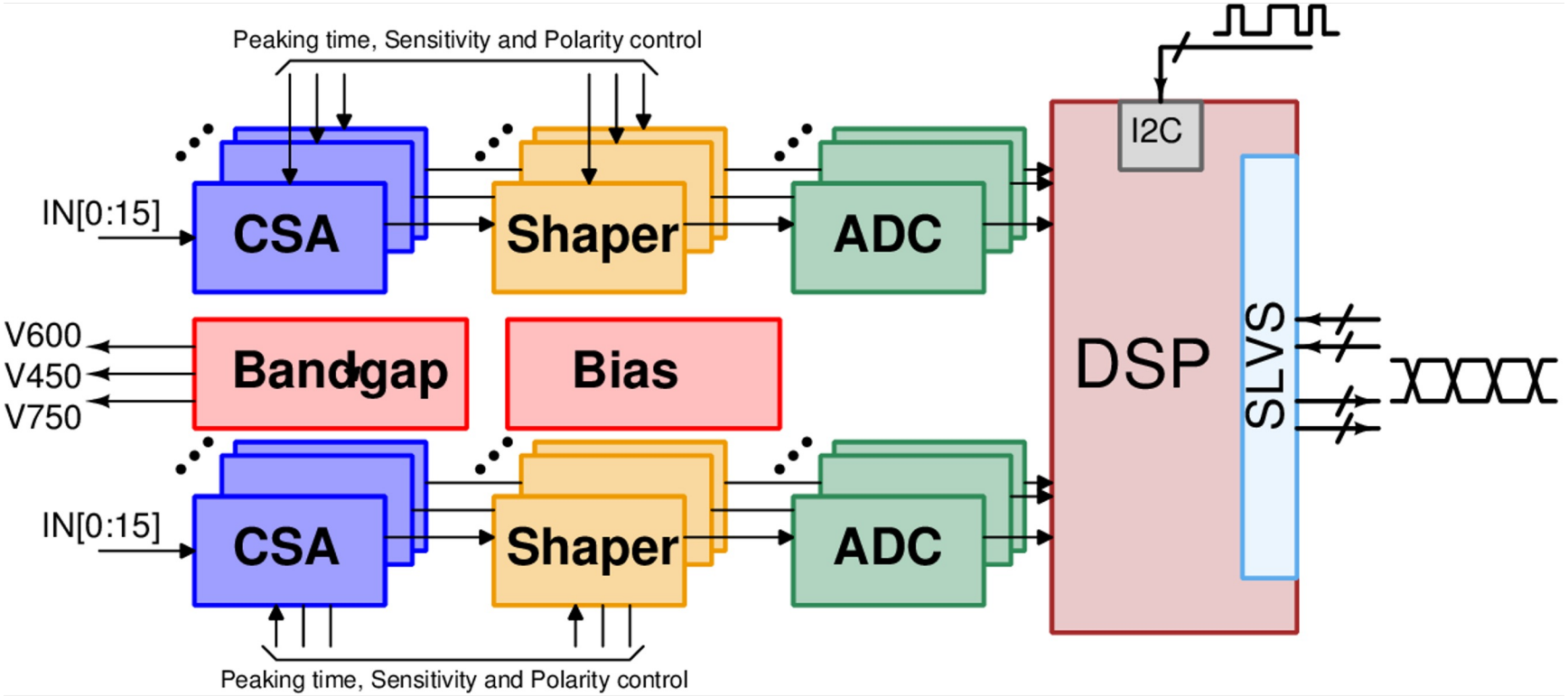
GEM MWPC

- TSMC CMOS 130 nm, 1.25V technology
- 32 channels, Front-end + ADC + DSP
- package size ≤15x15mm^2 (total footprint)
- ADC: 10-bit resolution, 10MS/s, ENOB>9.2
  (Alice TPC is eventually using: 5MS/s, to keep BW requirement in the readout chain lower)
- DSP functions: pedestal removal, baseline shift corrections, zero-suppression
- Data transmission: up to 11 e-link at 320 Mbps to GBTx, SLVS I/O
- Power < 32 mW/channel (Front End + ADC) V4, typical configuration, usually 20mW/ch or less.

TPC Mode	MCH Mode
▪ Negative Input charge ▪ Sensor capacitance: 12 – 25 pF ▪ Sensitivity: 20mV/fC & 30mV/fC ▪ Noise: ENC ≤ 580 e$^-$ @ 18.5pF ▪ Peaking time: ~160 ns ▪ Baseline return: <500 ns	▪ Positive input charge ▪ Sensor capacitance: 40–80 pF ▪ Sensitivity: 4mV/fC ▪ Noise: ENC ≤ 950 e- @ 40pF 1600 e- @80pF ▪ Peaking time: ~300 ns ▪ Baseline return: <550 ns

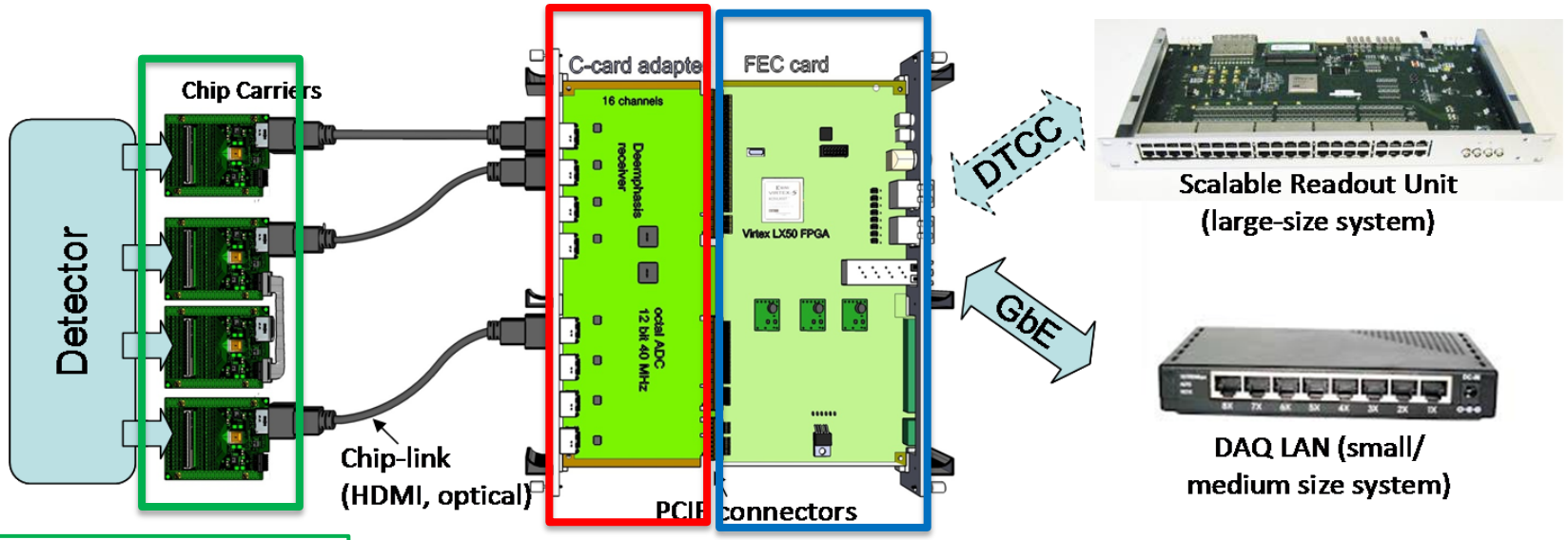
Later, a modified version with 80/160 ns shaping, 20/30 mV/fC gain, 20MSps optimized ADC, has been designed, fabricated, and validated. Presently available as well.

SAMPA Block Diagram



The SRS idea

crate close to the experimental apparatus

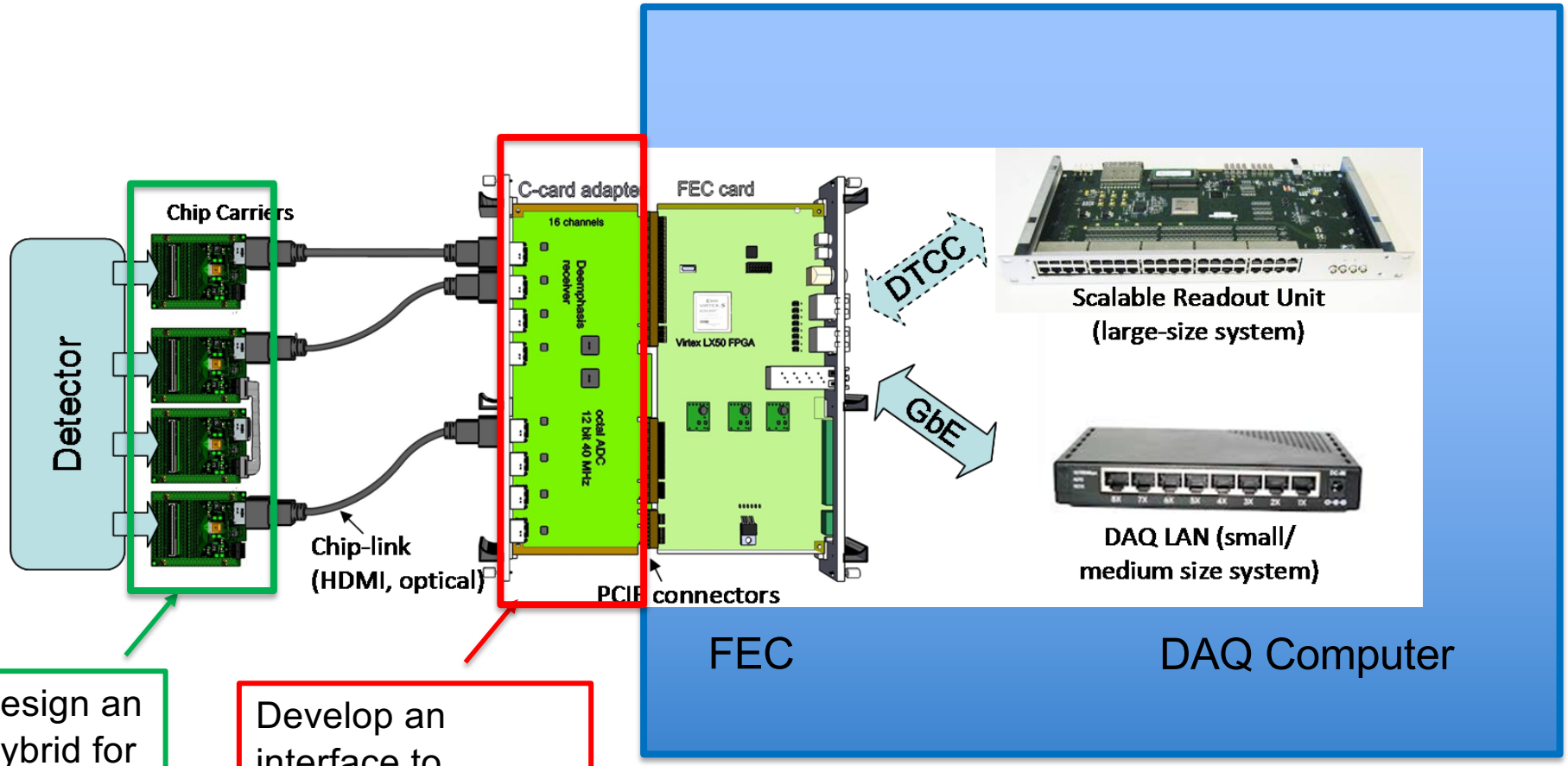


Front-end ASIC (close to detector)

An interface between ASIC and FEC card

FEC: handle the data taking of a set of ASIC. Handle the communication with the Readout Unit / DAQ

What's needed for SAMPA in SRS



Design an Hybrid for SAMPA

Develop an interface to manage the communication with SAMPA

KEEP/reuse, but FEC FPGA firmware

Acquisition software needs to be ~~adapted!~~

developed as well

Design strategy

- Hybrid compatible with present SRS standards
  - HRS input connector
  - Reading 128 Chs => 4 SAMPAs
  - "Slim" (two hybrids should be mounted side by side on the detector)
    - SAMPA is packaged in 15x15 mm^2 BGA, max 2 SAMPA side by side
    - Hybrid "almost" specular double face, top side reads channels 1-32/65-96, bottom side (HRS side) reads 33-64/97-128
- Handling, at least partially, the SAMPA output bandwidth
- Avoiding FPGA in the hybrid
- Lower voltage power connector for SAMPA, then each power domain served by dedicated LDO
  - SAMPA-analog
  - SAMPA-digital
  - SAMPA-ADC-reference
- Second power connector for the auxiliaries elements on the hybrids

Output connector (why DP)

- SRS uses HDMI cables, 19 pins
- Hybrid connects with adapter card in the crate, which is also ASIC specific
- Chose to go for DP port (20 pins), to route so all needed signals
- 4 DP main lanes used to transport output data
  - SAMPA uses up to 11 SLVS link at 320 Mbps
    - This signal cannot travel over long distances… should be converted in any case
  - Use a Serializer to convert 4 SLVS @320 Mbps to 1 LVDS lane@1280 Mbps
  - DP supports >2Gbps/lane already in it version 1.0 (so any DP cable does)
  - present prototype (v1): each SAMPA has a dedicated lane (initial idea)
  - protype V2, experiment new grouping
    - all the 4 links"1" of the 4 SAMPAS go to Lane_1
    - all the 4 links"2" of the 4 SAMPAS go to Lane_2
    - etc.
- Remaining DP pins used to transfer triggers and I2C signals

Output connector (why DP)

- SRS uses HDMI cables, 19 pins
- Hybrid connects with adapter card in the crate, which is also ASIC specific
- Chose to go for DP port (20 pins), to route so all needed signals
- 4 DP main lanes used to transport output data
  - SAMPA uses up to 11 SLVS link at 320 Mbps
    - This signal cannot travel over long distances… should be converted in any case
  - Use a Serializer to convert 4 SLVS @320 Mbps to 1 LVDS lane@1280 Mbps
  - DP supports >2Gbps/lane already in it version 1.0 (so any DP cable does)
  - present prototype (v1): each SAMPA has a dedicated lane (initial idea)
  - protype V2, experiment new grouping
    - all the 4 links"1" of the 4 SAMPAS go to Lane_1
    - all the 4 links"2" of the 4 SAMPAS go to Lane_2
    - etc.

  if it does work, moving to HDMI (max 3 lanes) would be possible, as well as "daisy chaininig" of 2 (both connectors) or 3(HDMI)/4(DP) hybrids

- Remaining DP pins used to transfer triggers and I2C signals

Functionalities overview

DSP

Top Level Functionality

- 4 primary filter blocks
  - Individual correction per channel
  - Baseline correction
    - 1 FIR filter
    - 1 Slope based filter
    - 1 IIR filter
    - Lookup table correction(Pedestal Memory) f(t);f(din)
    - Conversion f(din)
    - Fixed correction
  - Tail cancellation
    - 1 IIR filter
- Configuration
  - Configurable through I2C
  - 1 global register unit, 32 sets of channel registers
- Radiation tolerant
  - TMR on almost all flip-flops
    - except on part of data path
  - Hamming protected headers
- Compression
  - Zero suppression with run length encoding
    - Forward linked list for easier decoding
  - Cluster sum
    - Uses zero suppression with run length encoding , but sums cluster into 20bit word
  - Huffman
    - Differential encoded data
    - Programmable table of codes for +17 to -17
    - Values outside table have special Huffman code prepended to raw 10bit value
- Design for test
  - JTAG boundary scan
  - Built in memory tester
  - Scan chain (on >98% of digital block flops)

    (not implemented at hybrid level)

(not implemented at hybrid level)

Readout

- Selectable number of serial links up to 11
  - 320/160/80Mbps **up to 4 links in the hybrid**
  - Channels divided among links, no load sharing
  - Which channel goes to which link and in which order can be selected
  - Data is packet based (header + payload)
    - One packet per channel per event
- Event modes
  - Triggered
  - Continuous
  - Selectable event length up to 1024 samples
  - up to 192 pre-trigger samples “backshift”
- Event buffer per channel
  - 6144(6K) words of compressed samples
  - 256 words of headers
  - Header still created if data memory goes full

- Daisy chain between chips
  - Multiple devices can share a single serial link to readout unit
  - 2K word buffer in the receiving side
- Direct ADC serialization
  - Data serialized directly from ADC at 32xADC speed over 10 links
  - Raw data, no filtering, no headers
  - Sync pattern on startup, receiver should maintain sync after that
  - 2 modes
    - 10 bits is sent consecutively for channel 0-31 each 32xADC cycle
    - 5 lower bits, then 5 higher bits consecutively for channel 0-15 is sent on link 0-4 and for channel 16-31 on link 5-9
  - Clockgate the rest of the system to save power

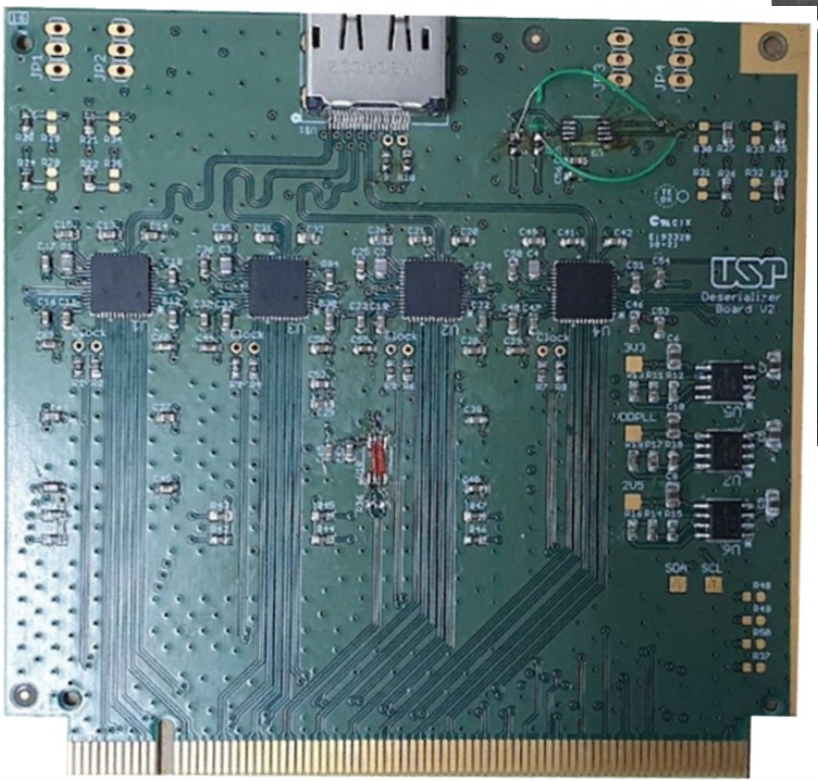
(not implemented at hybrid level)

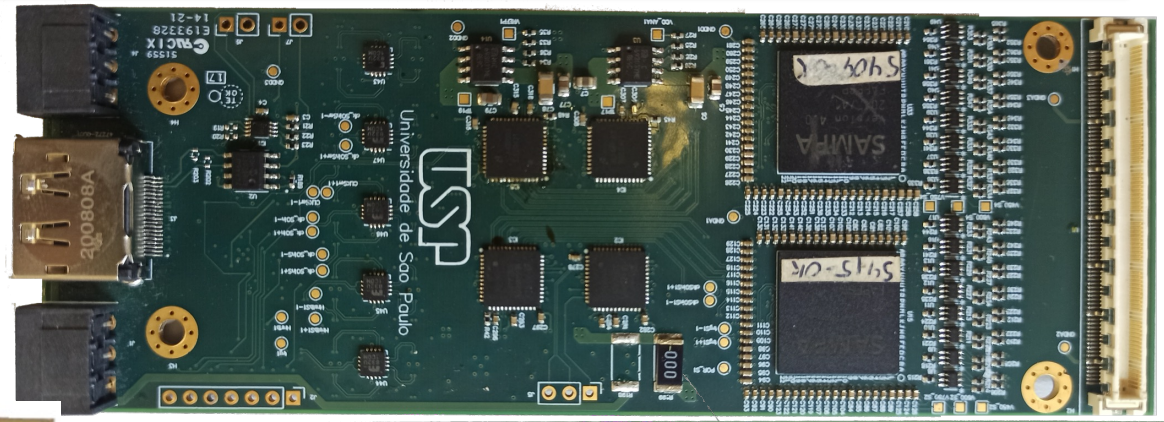
Present Status

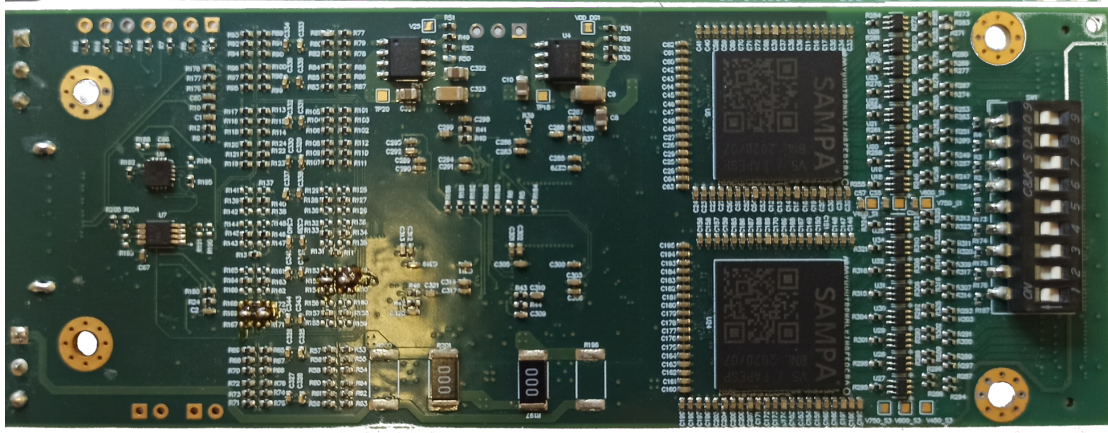
- After the first round (hybrid_v0), minor bugs were fixed and a fully functional prototype of Hybrid (v1) and Adapter_board were fabricated
- FEE_v6 FPGA firmware developed to handle configuration (I2C), receiving the data flux, and send via TCP/IP
- Proper software to receive the data, to visualize them on the fly (monitor) and (eventually) to produce root file developed and working

Adapter board & Hybrid

Adapter board

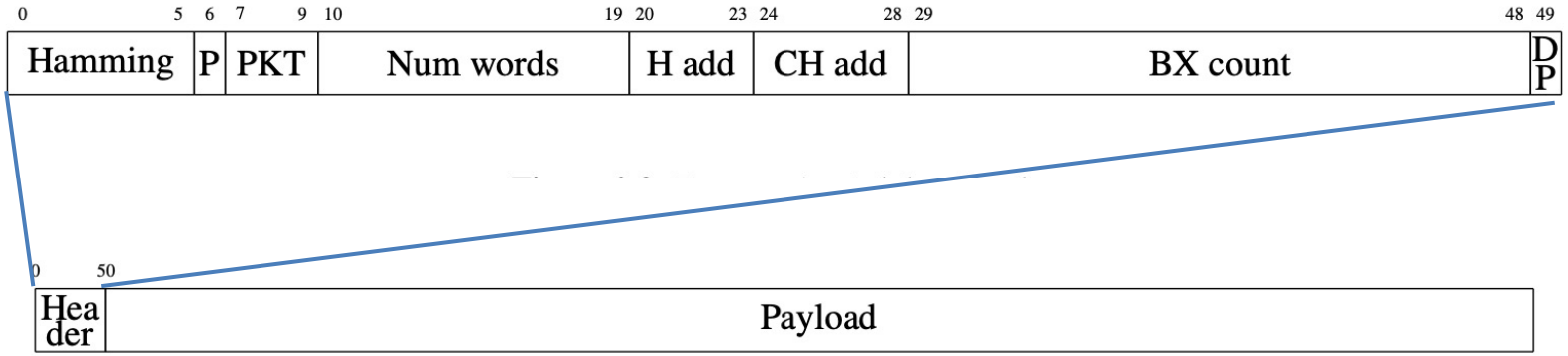


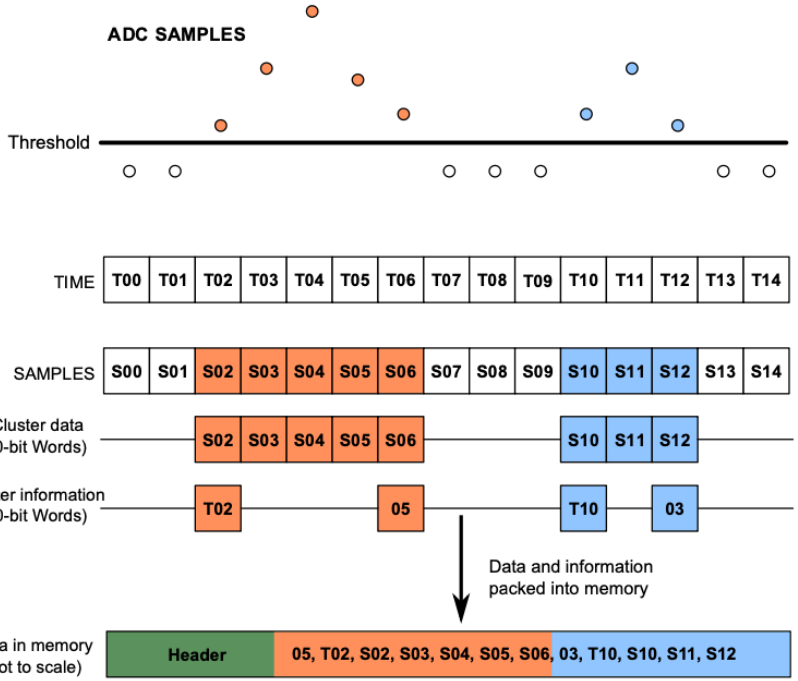




Hybrid_v1
(50 mm width compliant)

SAMPA data format





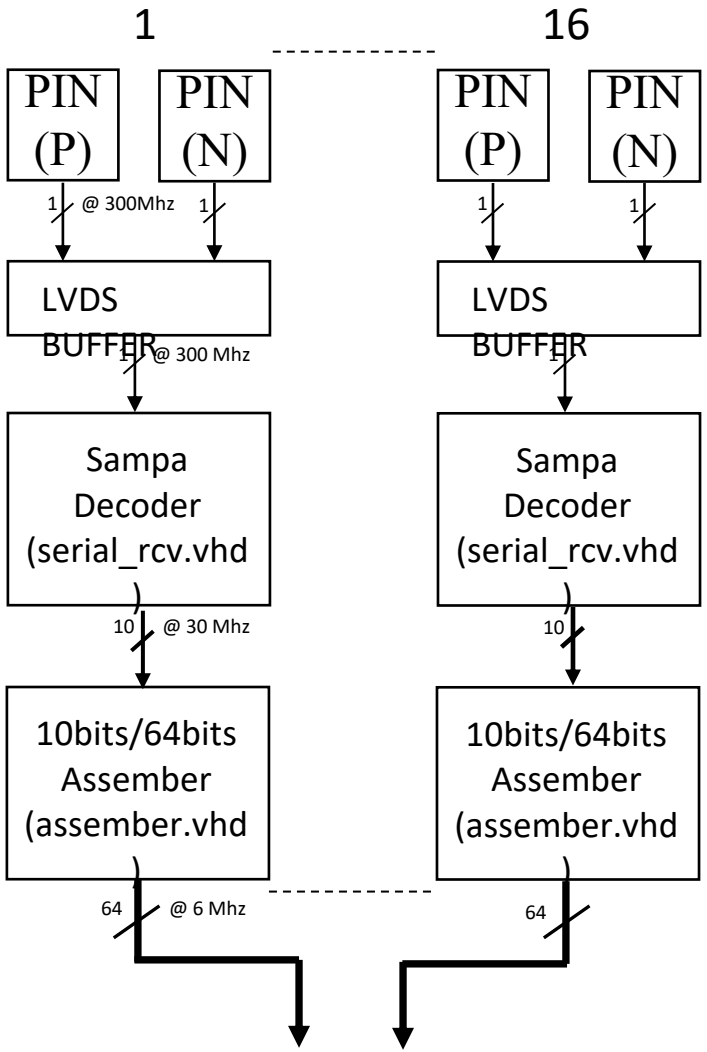
Payload (ZS case)

Name	Bits	Description
Hamming	6	Hamming code
P	1	Parity (odd) of header including hamming
PKT	3	Packet type, see table 2.6
Num words	10	Number of 10 bit words in data payload
H add	4	Hardware address of chip
CH add	5	Channel address
BX count	20	Bunch-crossing counter (40MHz counter)
DP	1	Parity (odd) of data payload

Header

Package coding on FEC

for each event, each SAMPA prepares a package Header+Data for each of its channels. Then the complete packages, are sent, channel by channel, via eLink.

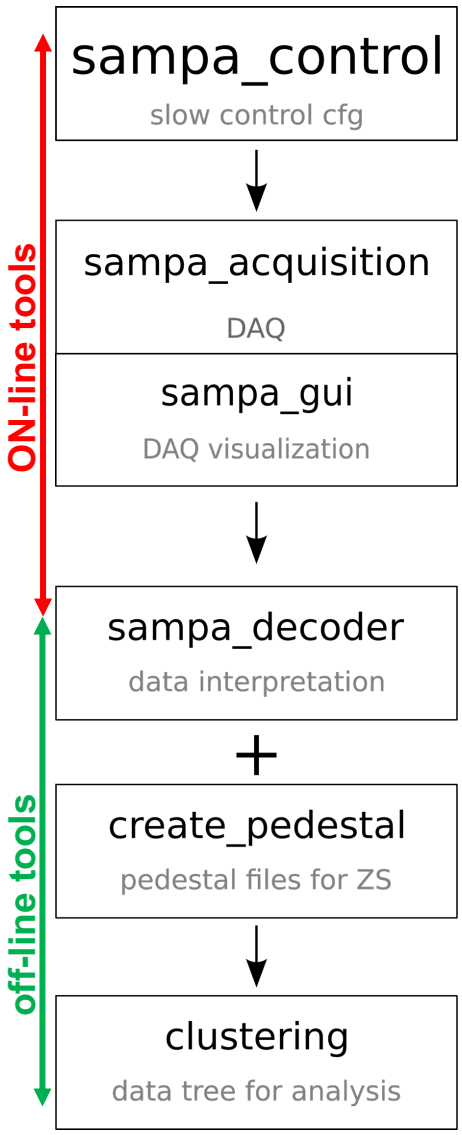


there are (up to) 16 data link (FIFO) input into the FEC, only 1 TPC/IP output.
It is needed to code SAMPA package + FIFO information

64-bit words

- 0-9 – 10 bits (fist data word, 10 bits)
- 10-19 - 10 bits (second data word, 10 bits)
- 20-29 - 10 bits (third data word, 10 bits)
- 30 – not used/defined
- 31 – 1 bit, present queue (FIFO) full
- 32-43 - 10 bits (fourth data word, 10 bits)
- 44-52 - 10 bits (fifth data word, 10 bits)
- 52-61 - for FIFO (queue) identification
- 62-63 - 2 bits : for coding the kind of data:
  01 Data; 10 *Header;* 11 *Trigger too early*

SAMPA Readout SW and tools



[I2C communication] **sampa_control:** request/reply protocol communication between the computer and the FEC/SAMPAs to configure the acquisition (start/stop, internal/external trigger, waveform length, latency, zero suppression threshold) [I2C communication]

[TCP/IP receiver] **sampa_acquisition/sampa_gui:** a tool that capture the packages from the network. Data is save still in "raw" format.

[On-line Monitoring] **sampa_gui** If the user wants to have a real time preview of the acquisition a graphic user interface process and decodes a small amount of data to create few histograms.

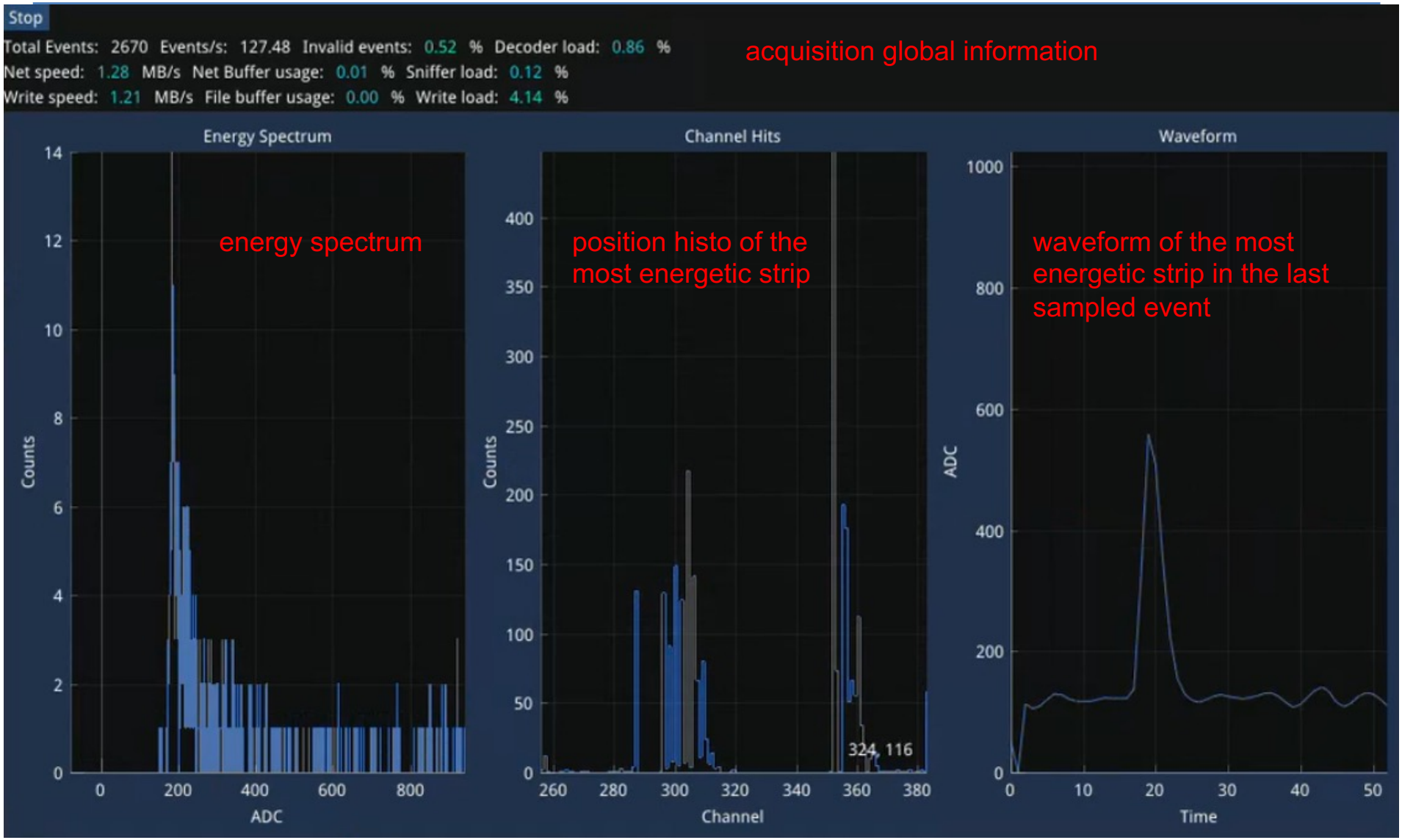
used here as well,

[DATA "Decoding"] **sampa_decoder:** the FEC built packages of 64 words which contains extra information and flags to identify "event" and chip/ch. The original data is divided in "chunks" to be sent. This block decodes the hexadecimal stream and extract the chunks of data, re-organizing it by event/chip/ch. i.e. transform the raw data produced by SAMPA+FEC (unordered hex) into a "human intelligible" root TTree. The event built contains information of time, channel and the complete waveform produced after a trigger.

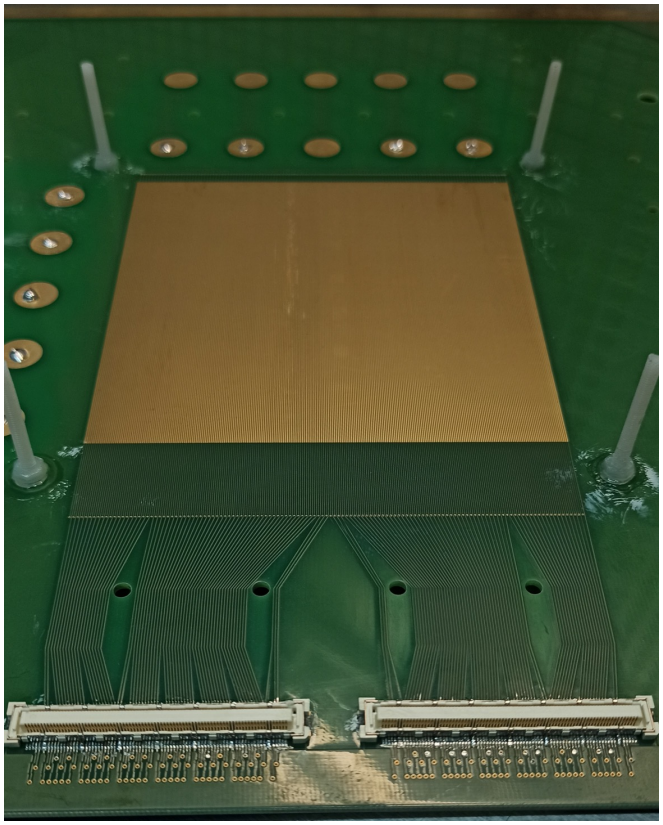
[Baselines] **create_pedestal:** process and produces a pedestal file containing information of the baseline levels and noise for each channel. Can be used for offline zero suppression or as input for the online zero suppression of SAMPA (work in progress).

[Reconstruciton] **clustering:** using a pedestal file build the clusters using a clustering algorithm (different options) The new file contains information regarding the centre of mass position of the cluster, energy and also its size (number of strips).

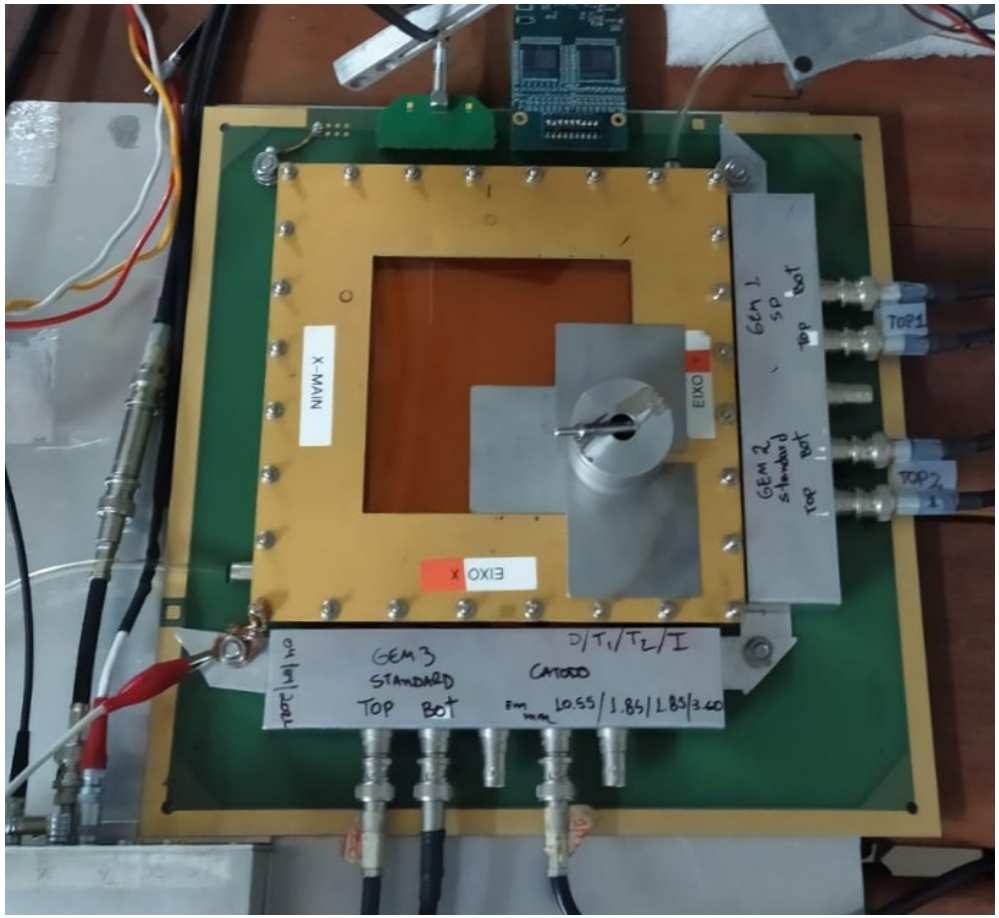
The GUI interface



The detector used to test the SAMPA-SRS-DAQ developed



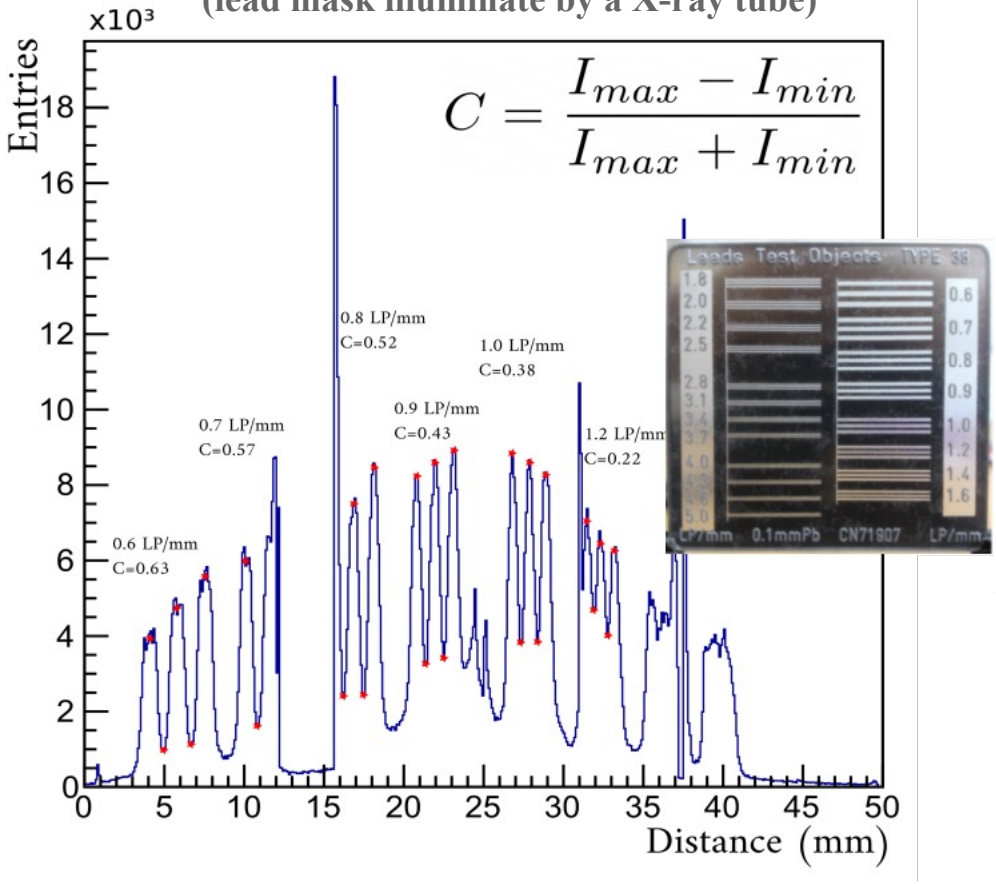
1D segmented readout: w. HRS connector
Strip width: 0.2 mm. Pitch: 0.4 mm. HRS

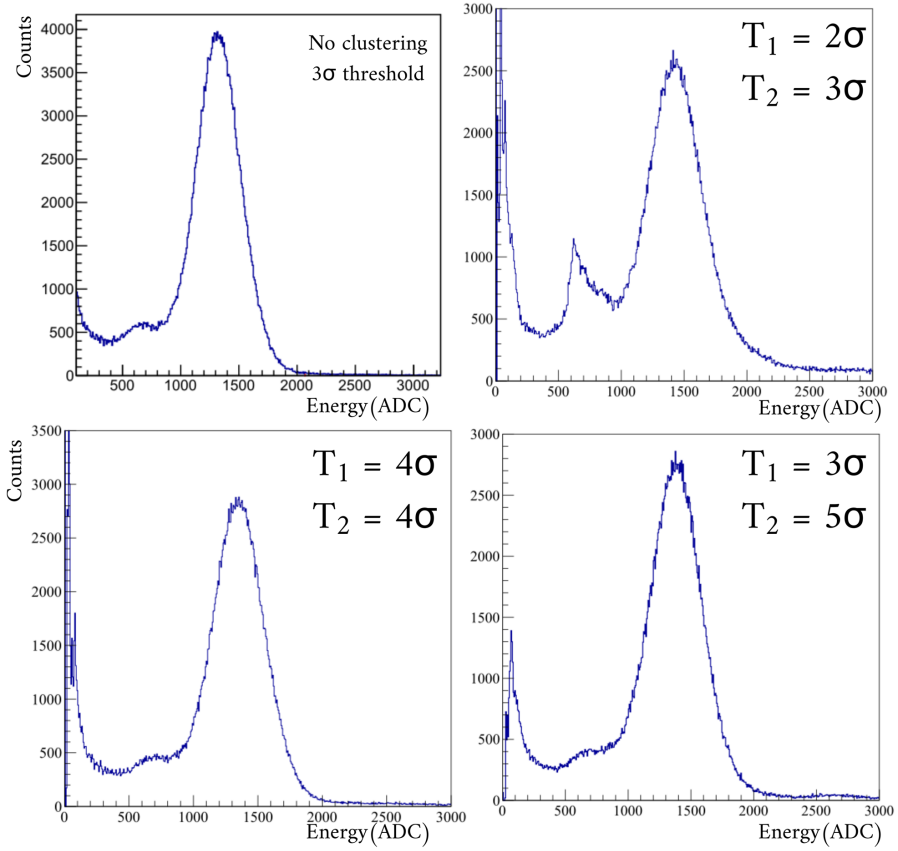


^{55}Fe source measurements

Results from the SAMPA-SRS-DAQ used to readout a real detector

Spatial resolution and contrast (lead mask illuminate by a X-ray tube)





^{55}Fe energy spectrum reconstructed with different thresholds for clustering (off-line)

Conclusions

A working prototype of acquisition system (including acquisition software tools) based on SAMPA on SRS has been developed.

The HW components:

- Hybrid Card with four SAMPAs, designed, fabricated and tested in his version V1
- Interface card (into the SRS crate) to demux the 4 higher speed links to the original 16 Sampa eLinks

The FEC FW

- coding the SAMPA data stream (16 links) to 1 TCP/IP stream

The SW (mainly in C++):

- Tool to handle I^2C communication
- “sniffer” to receive the TPC/IP package
- GUI with on-line (sampling) monitoring capability
- Decoder to produce organized Ttree form raw data