A FPGA-based digital readout system for a multi-channel X and γ-ray spectrometer

Riccardo Campana

INAF/IASF-Bologna

Workshop su applicazioni FPGA in ambito astrofisico

Pino Torinese (TO)

May 20, 2016

The XGS project

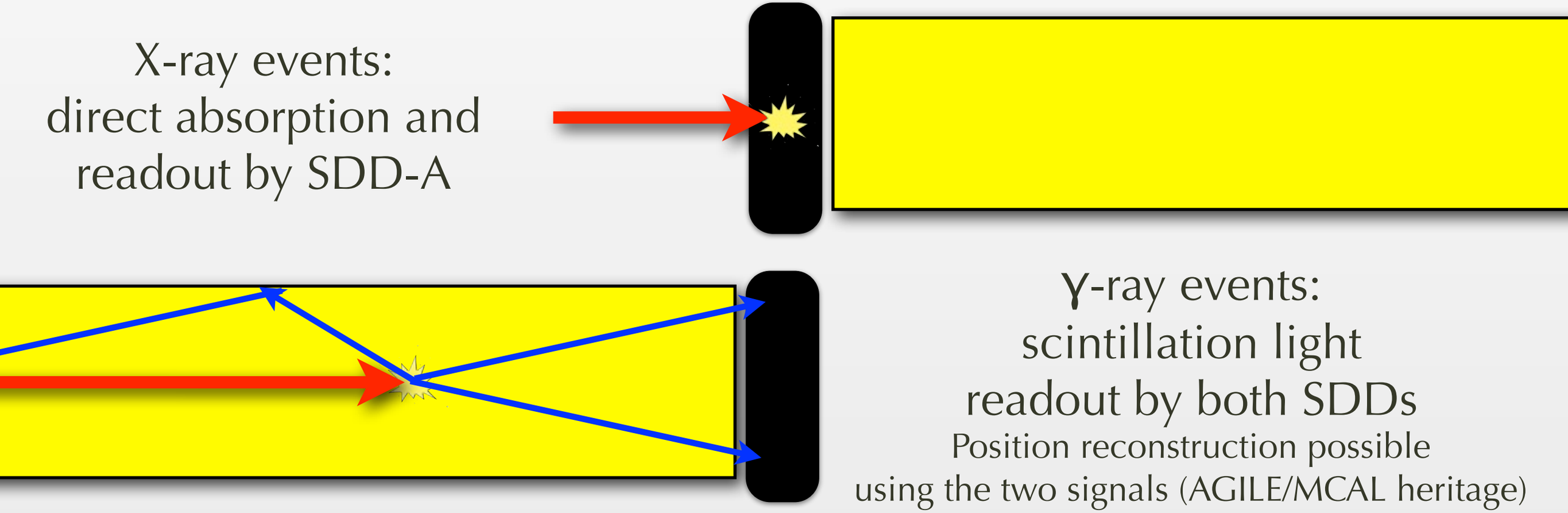
- **X and Gamma Spectrometer** (*one instrument to rule them all)*
- Based on **Silicon Drift Detectors** coupled with **scintillator** bars
- Huge sensitivity band: ~2 keV to 20 MeV
- Applications for space-based γ-ray astronomy and Earth observation:
  - ✓ THESEUS (ESA M4, M5)
  - ✓ ASTROGAM (ESA M4, M5)
  - ✓ PANGEA (ESA EE-9)
- Funded by INAF (TecnoPRIN 2014)

Architecture

Scintillator bar with double-SDD readout

Dual architecture (“Siswich” concept; Marisaldi et al. 2004, 2005)

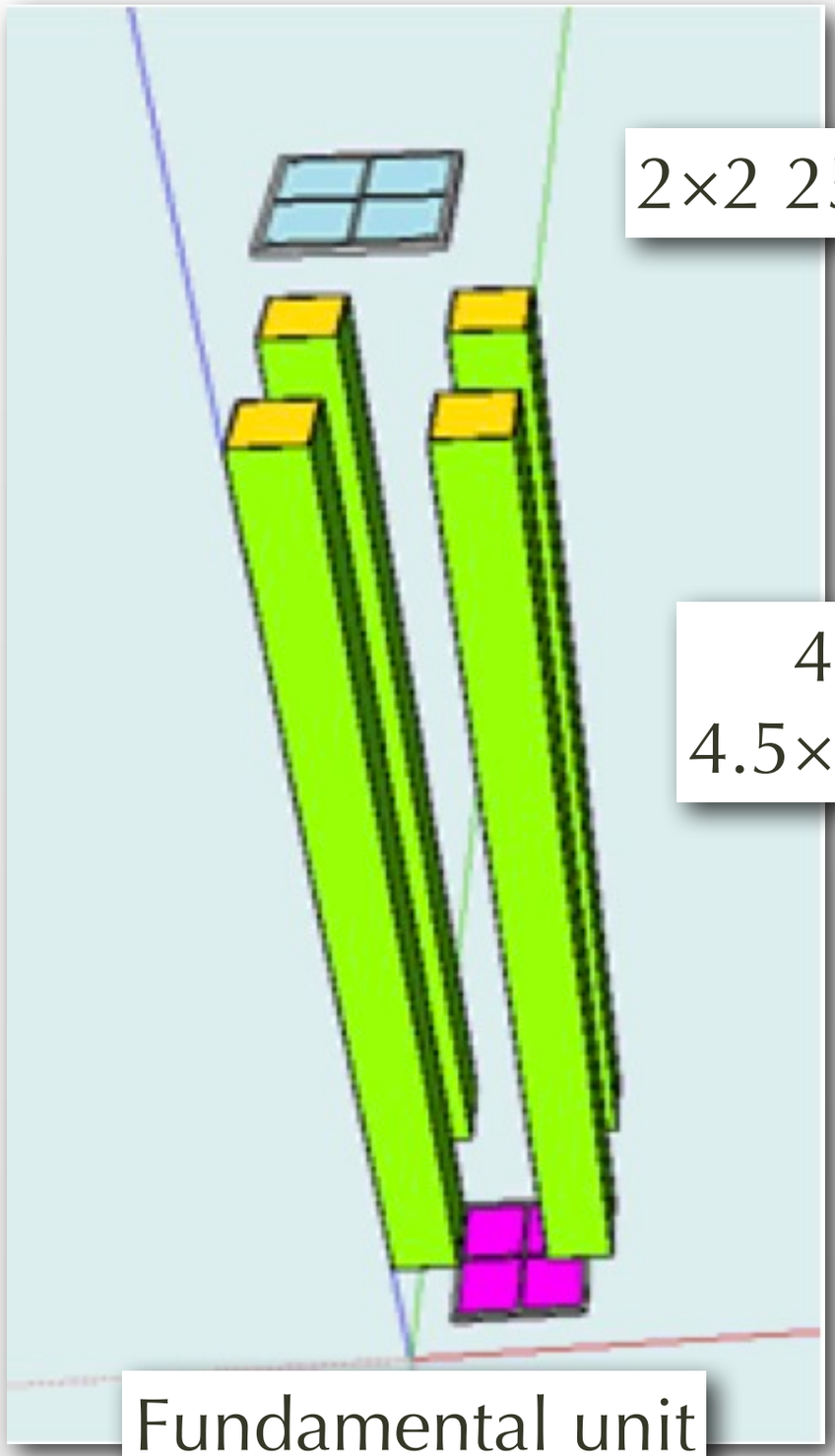
X-ray events:
direct absorption and
readout by SDD-A



γ-ray events:
scintillation light
readout by both SDDs

Position reconstruction possible
using the two signals (AGILE/MCAL heritage)

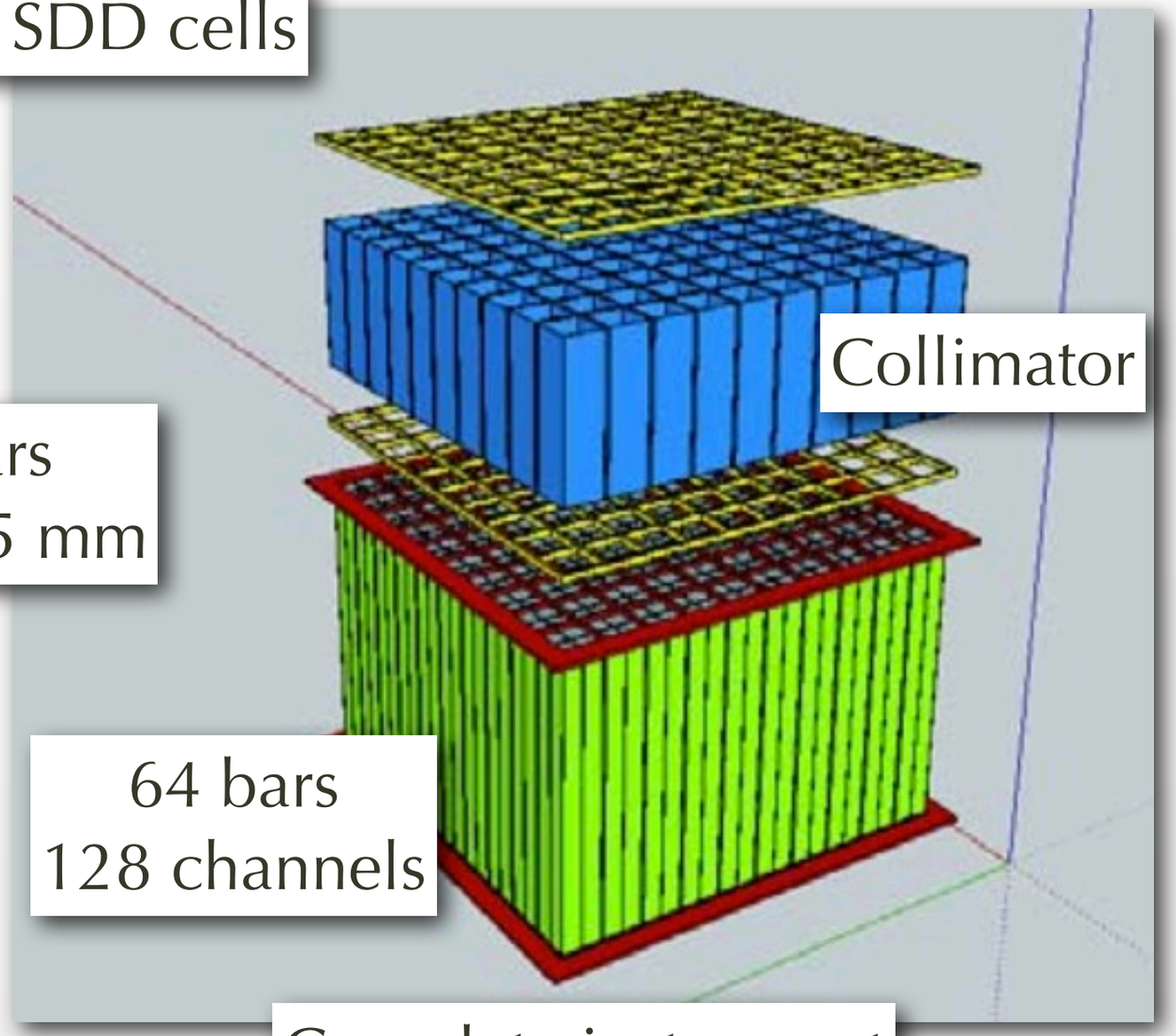
The XGS concept



2×2 25 mm² SDD cells

4 CsI bars
4.5×4.5×45 mm

Fundamental unit



Collimator

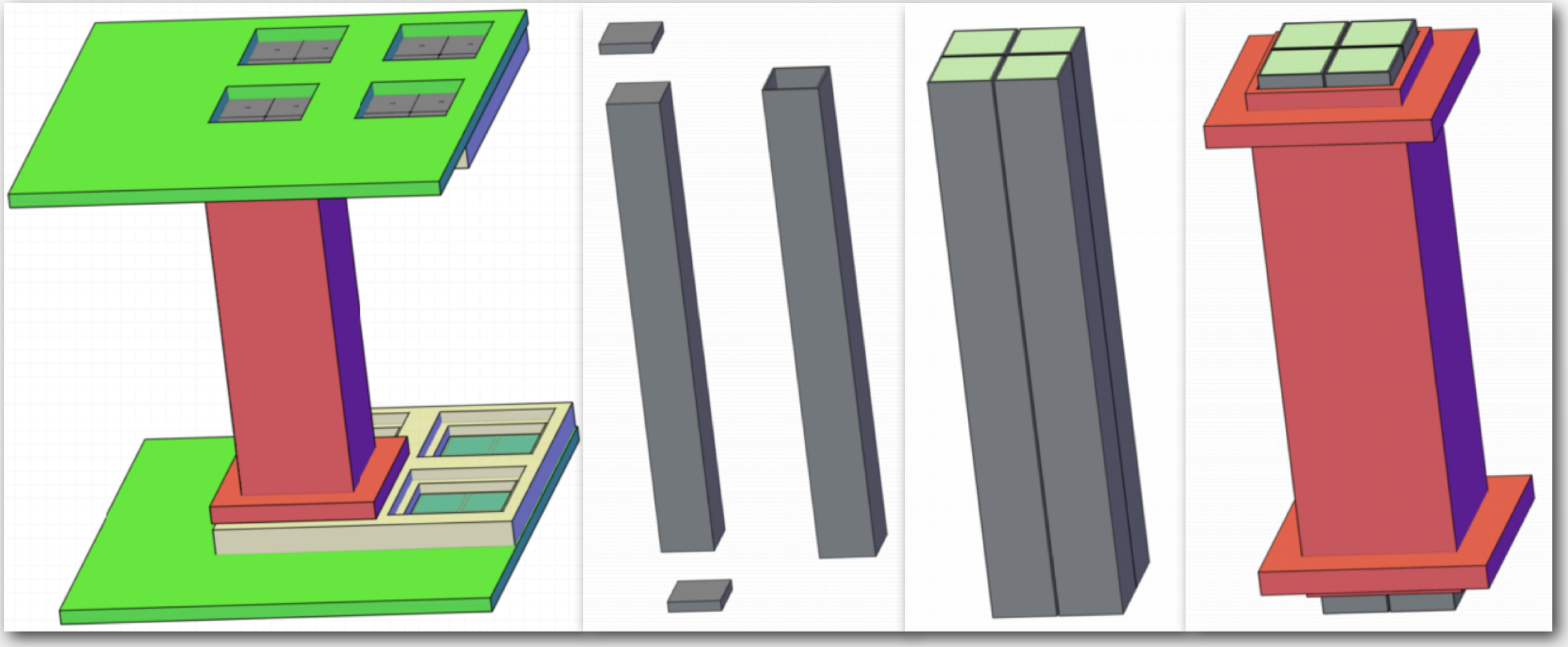
64 bars
128 channels

Complete instrument

The INAF-funded prototype

TecnoPRIN 2014 (IASF Bologna, IASF Milano)

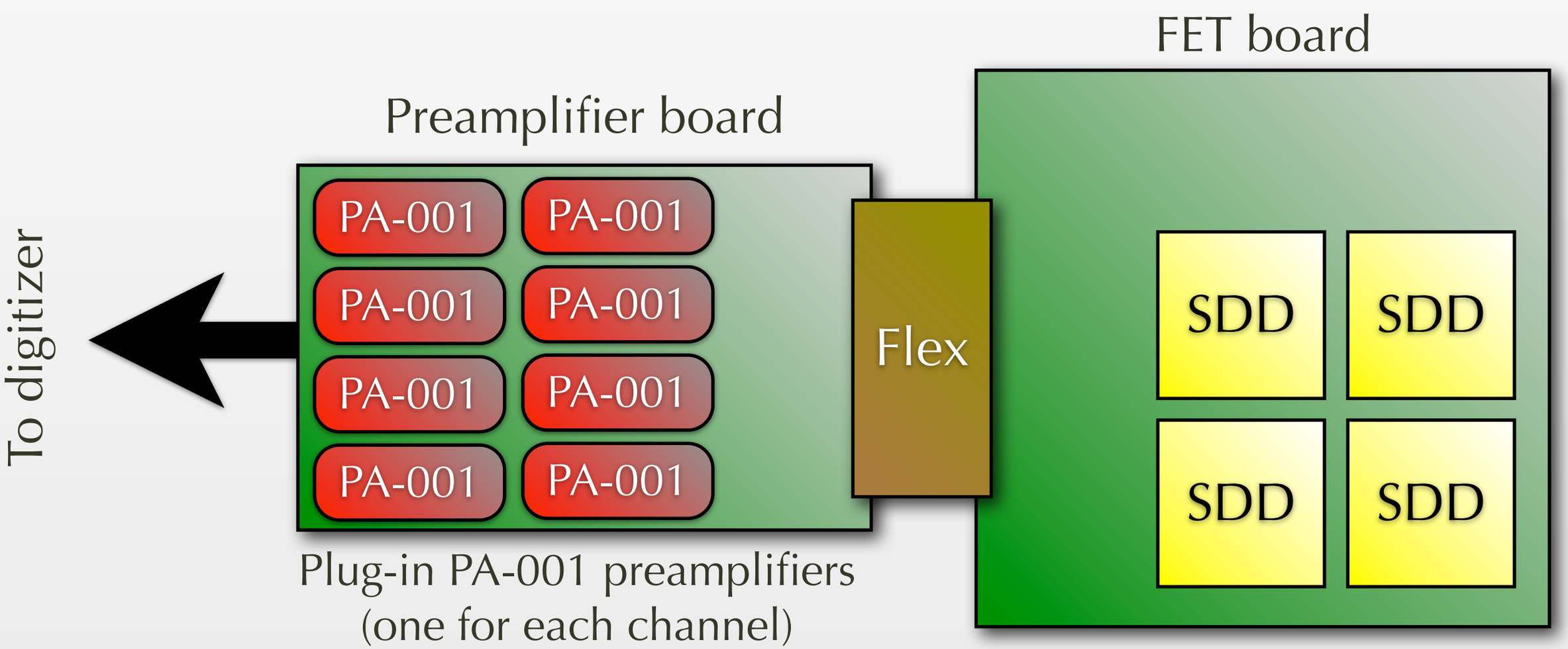
Up to 4 modules, each with 8×8 bars and 2×2 4-cell SDDs
FEE: discrete-electronics charge-sensitive preamplifier
+ digital signal processing



The INAF-funded prototype

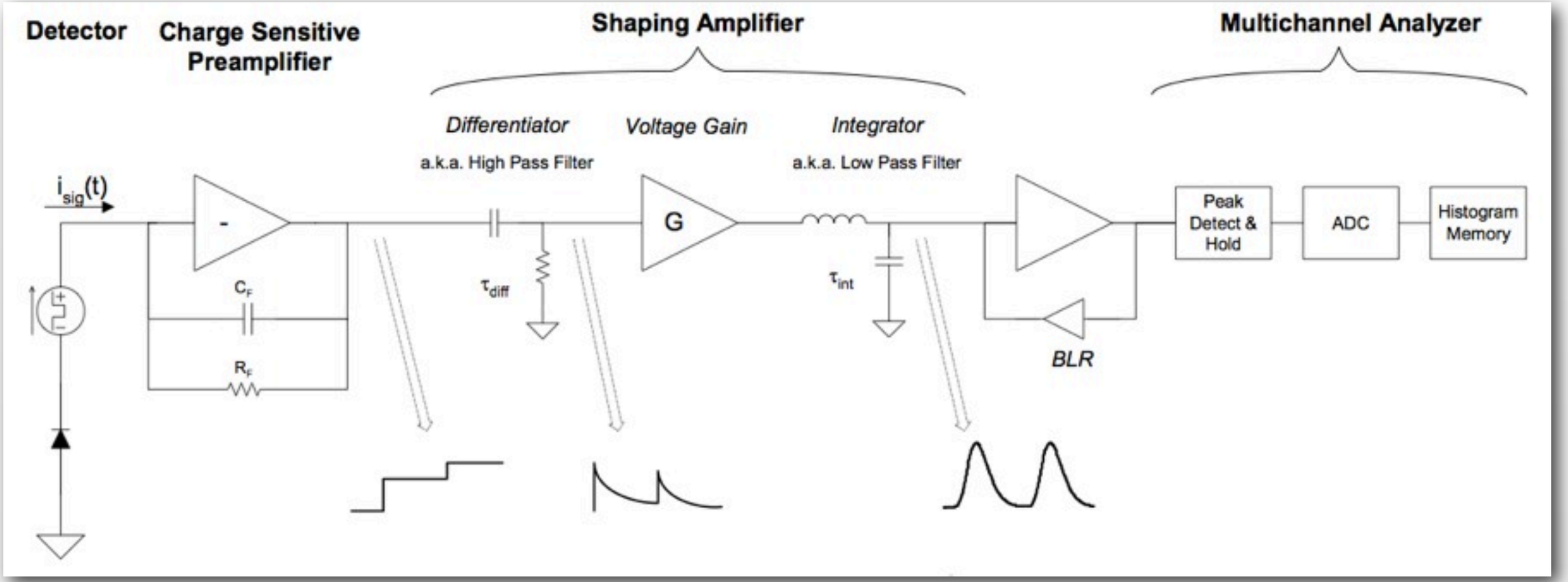
TecnoPRIN 2014 (IASF Bologna, IASF Milano)

Up to 4 modules, each with 8×8 bars and 2×2 4-cell SDDs
FEE: discrete-electronics charge-sensitive preamplifier
+ digital signal processing



Digital signal processing

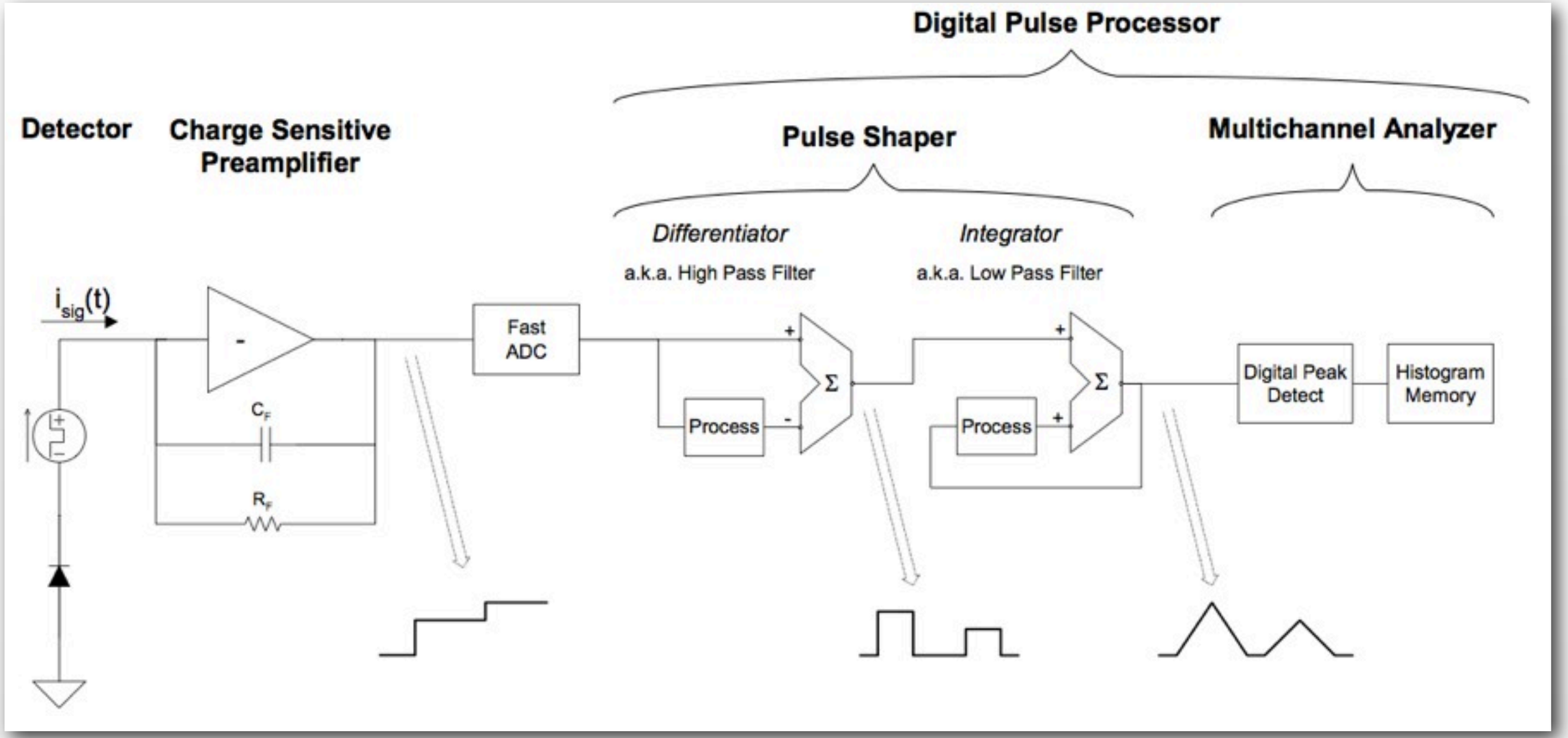
The classical analog acquisition chain



credits: http://amptek.com/pdf/dpp_theory.pdf

Digital signal processing

A digital acquisition chain



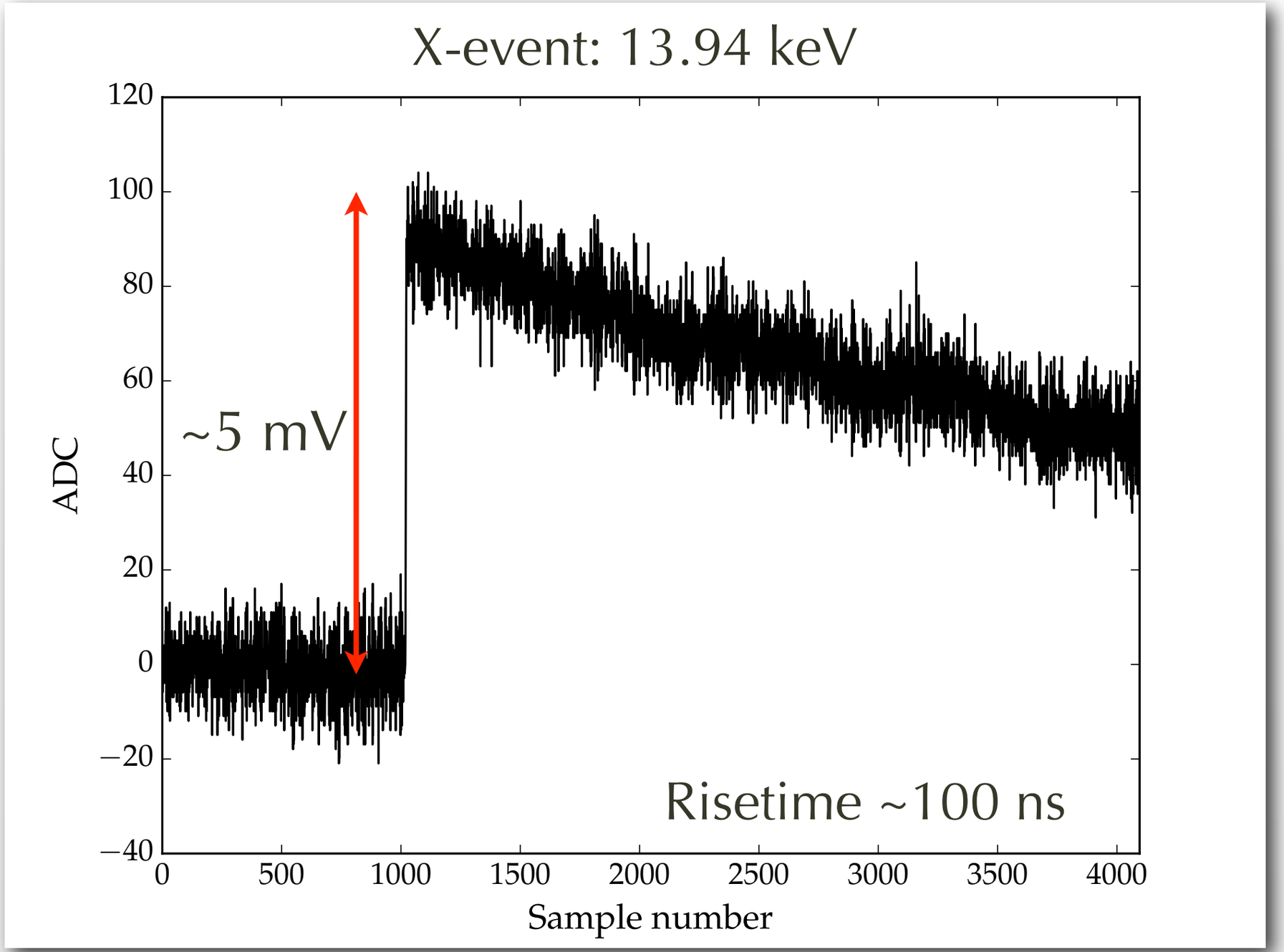
credits: http://amptek.com/pdf/dpp_theory.pdf

Digital signal processing

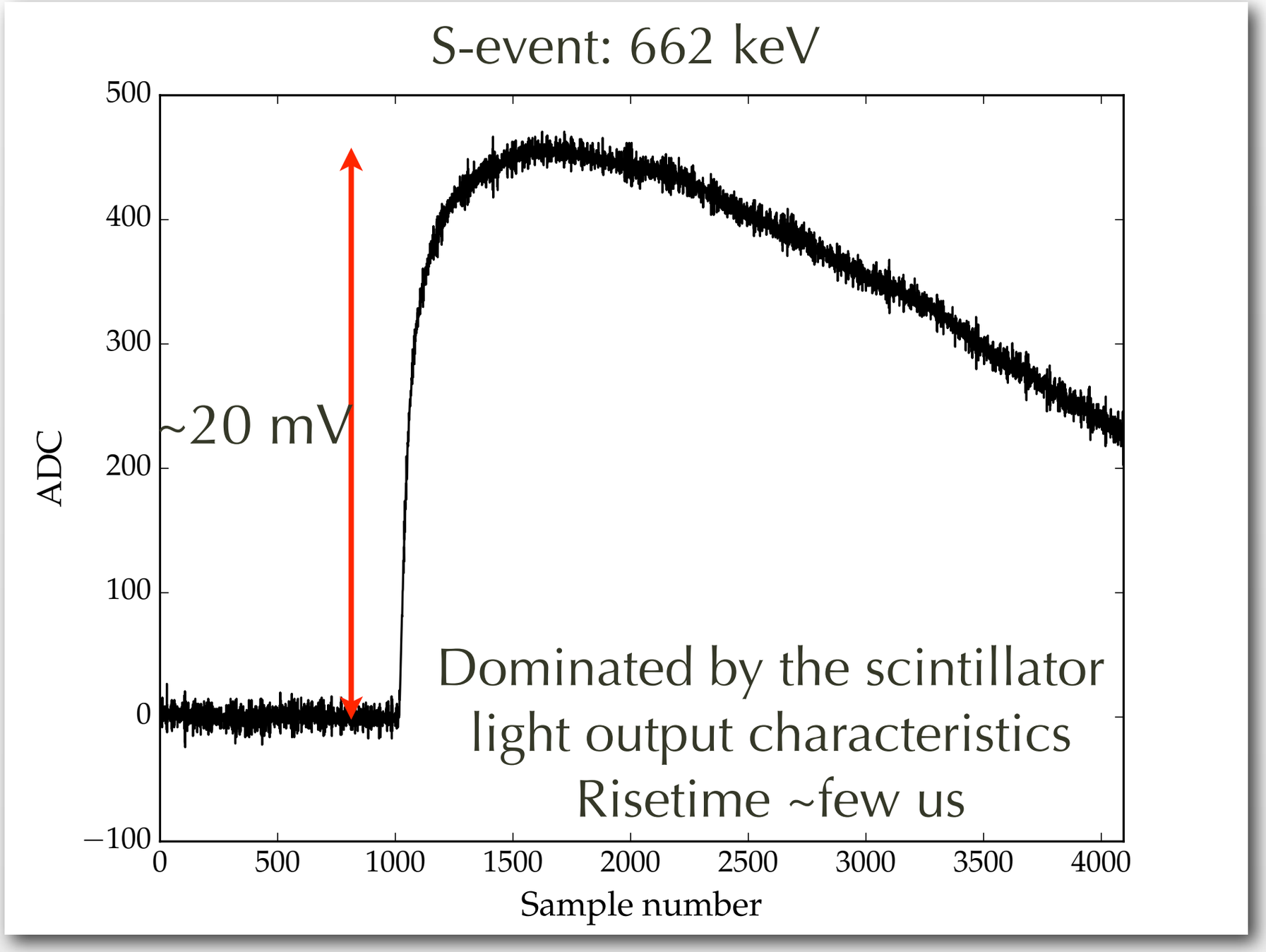
Digital signal processing has several advantages when developing a prototype:

1. Closely approximate the **ideal filter**
2. No dead time associated with the peak detect and digitization: **higher throughput**
3. *Analog system*: only **a few** shaping choices (shaping times & shapes).
   Digital system: **no limit** (also on **same data**)
4. Much higher stability and reproducibility

The actual signal



The actual signal

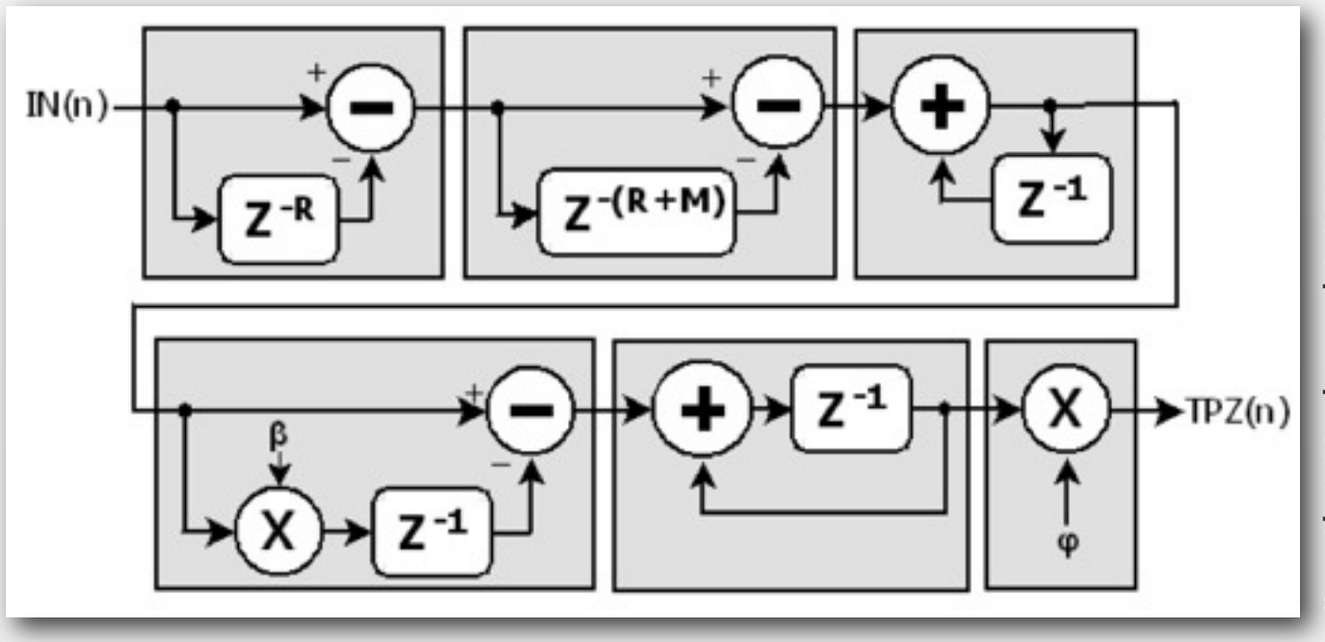


Trapezoidal filter

- Classical implementation of trapezoidal shaper (Jordanov & Knoll 1994, Guzik & Krakowski 2013)

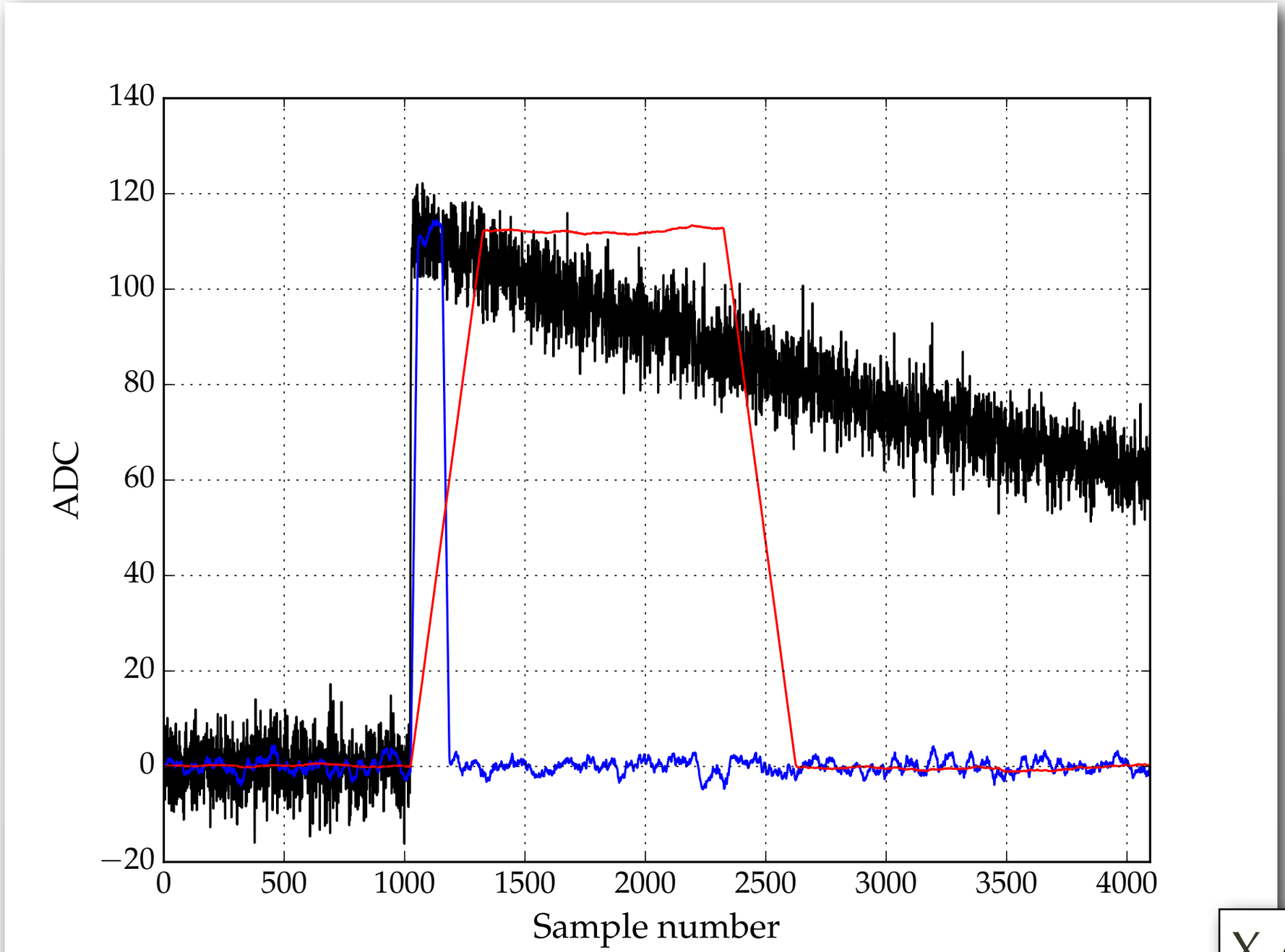
- Strictly valid for $\tau_{rise} << \tau_{fall}$ (negligible rise time)

- Easily implementable as a cascade of FIR and IIR filters



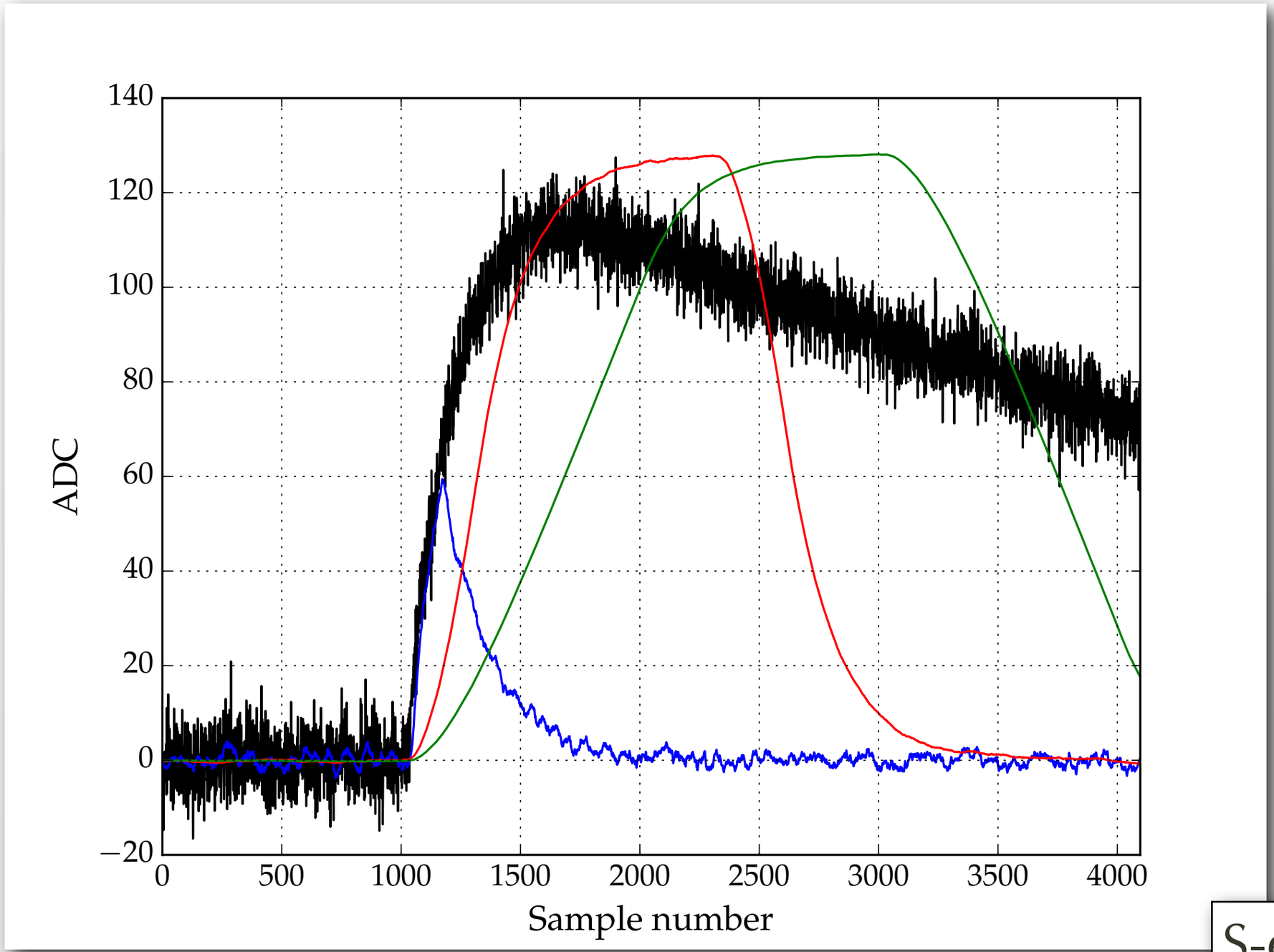
Guzik&Krakowski (2013)

Trapezoidal filter



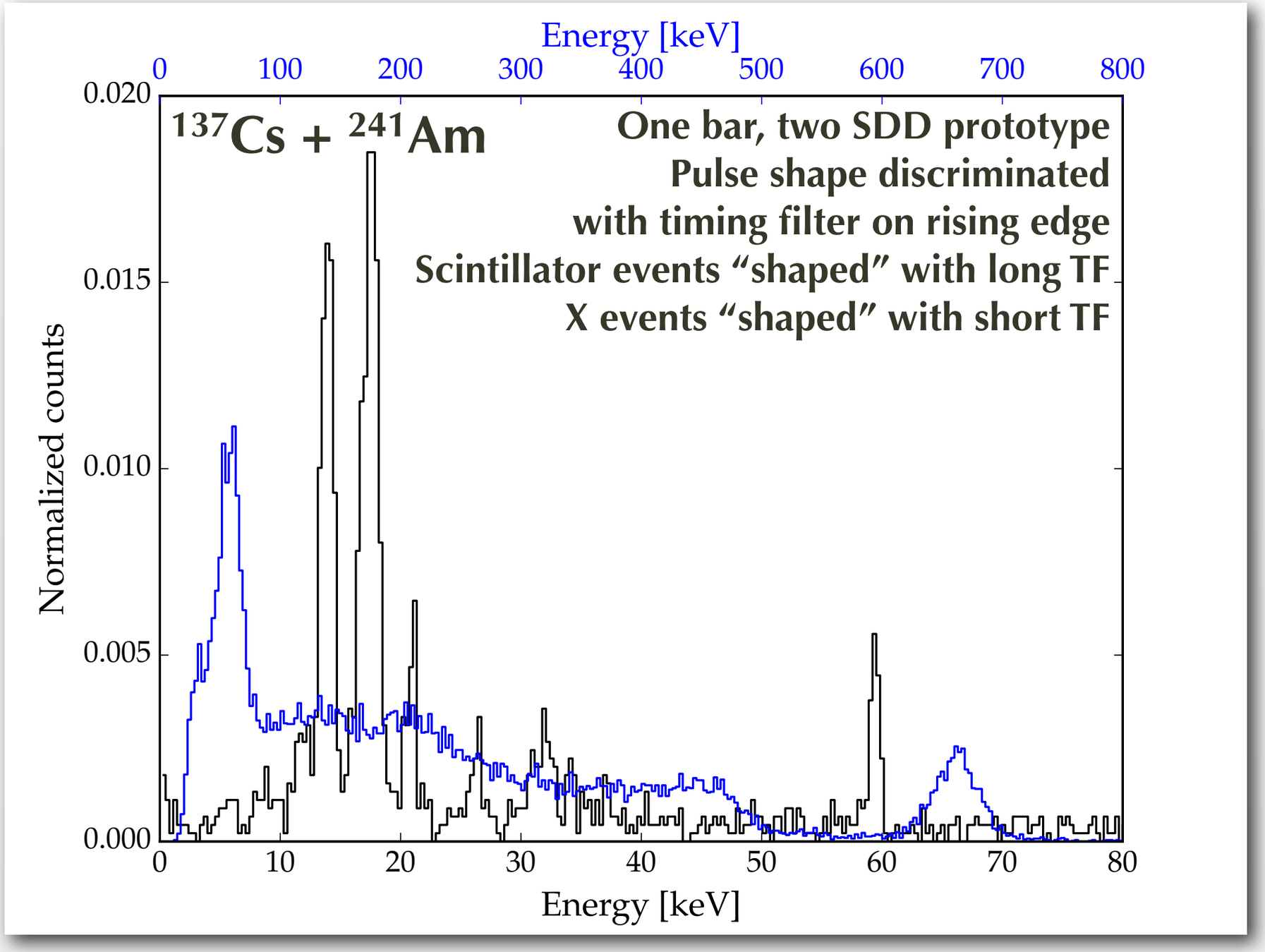
X-event

Trapezoidal filter



S-event

Example

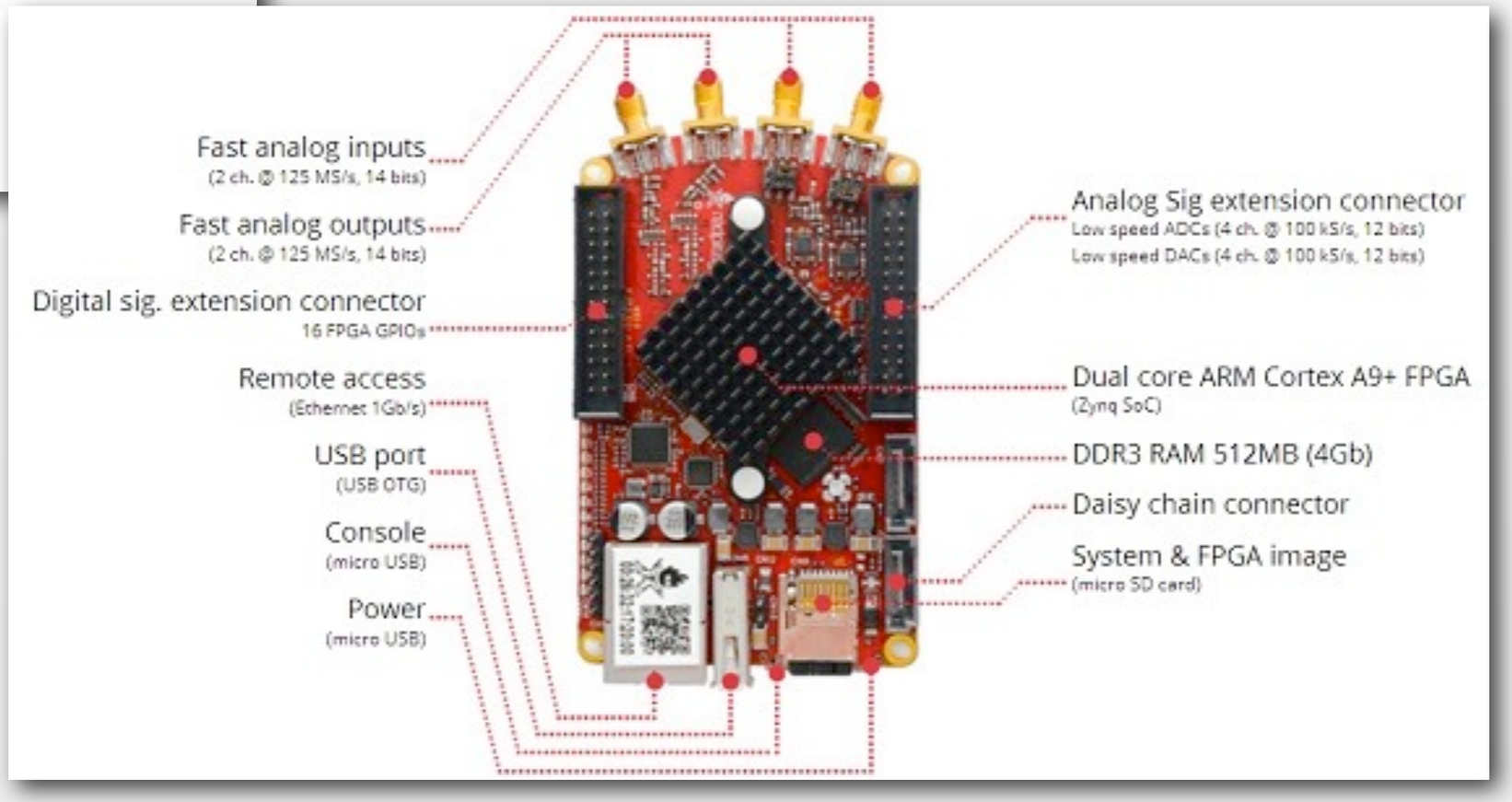

Energy [keV]
0
100
200
300
400
500
600
700
800
0.020
0.015
0.010
0.005
0.000
Normalized counts
^{137}Cs + ^{241}Am
One bar, two SDD prototype
Pulse shape discriminated
with timing filter on rising edge
Scintillator events “shaped” with long TF
X events “shaped” with short TF
0
10
20
30
40
50
60
70
80
Energy [keV]

DSP with RedPitaya

www.redpitaya.com

Low cost DSP board
(~200 €/board)
SoC Xilinx Zynq 7010
FPGA+ARM Cortex A9 CPU
Linux SO
FPGA source code available

- 2 input analog channels
- 2 output analog channels
- 14 bit ADC
- 125 MS/s + decimation
- GPIO pins
- 16384 samples buffers
- Ethernet output

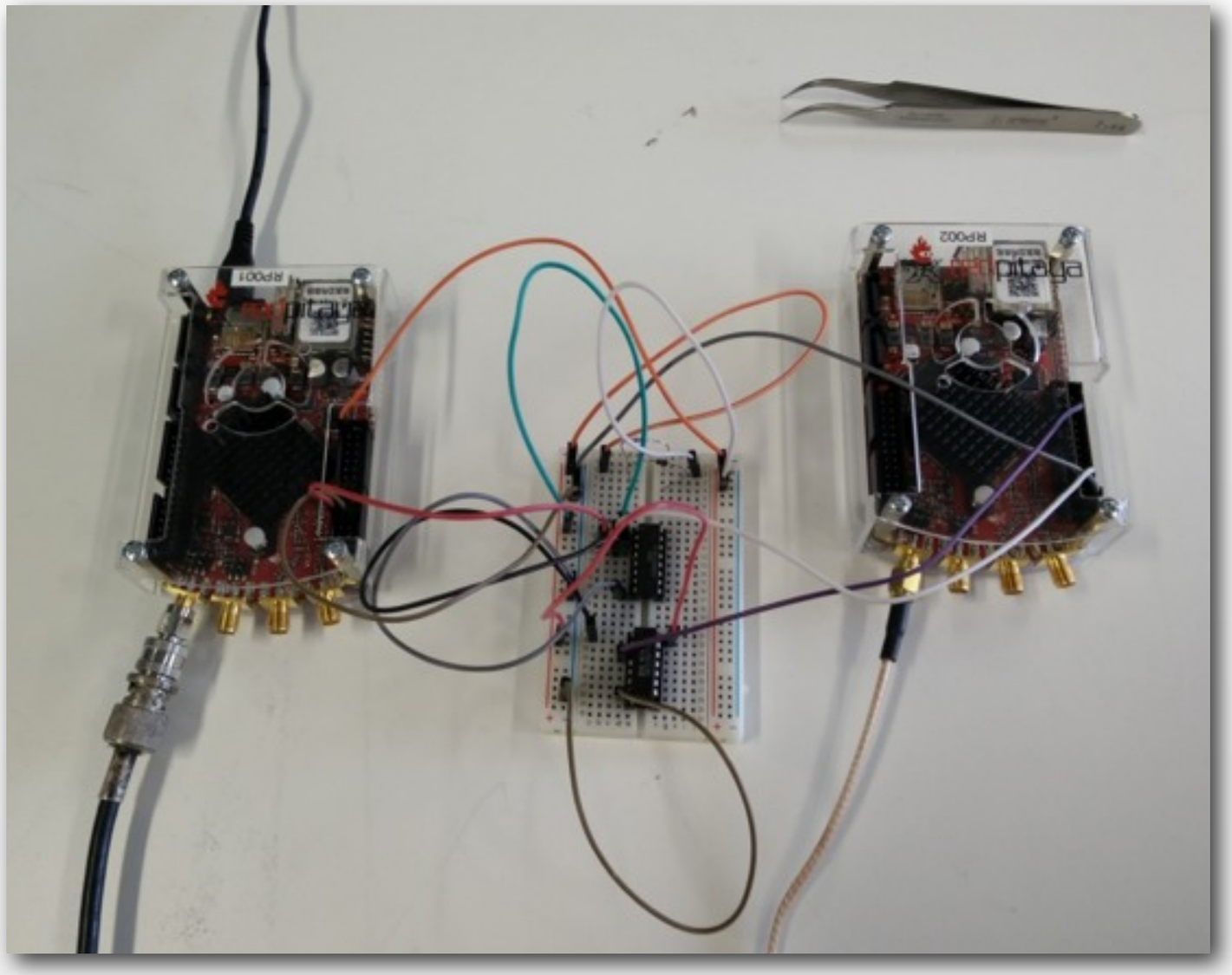


DSP with RedPitaya

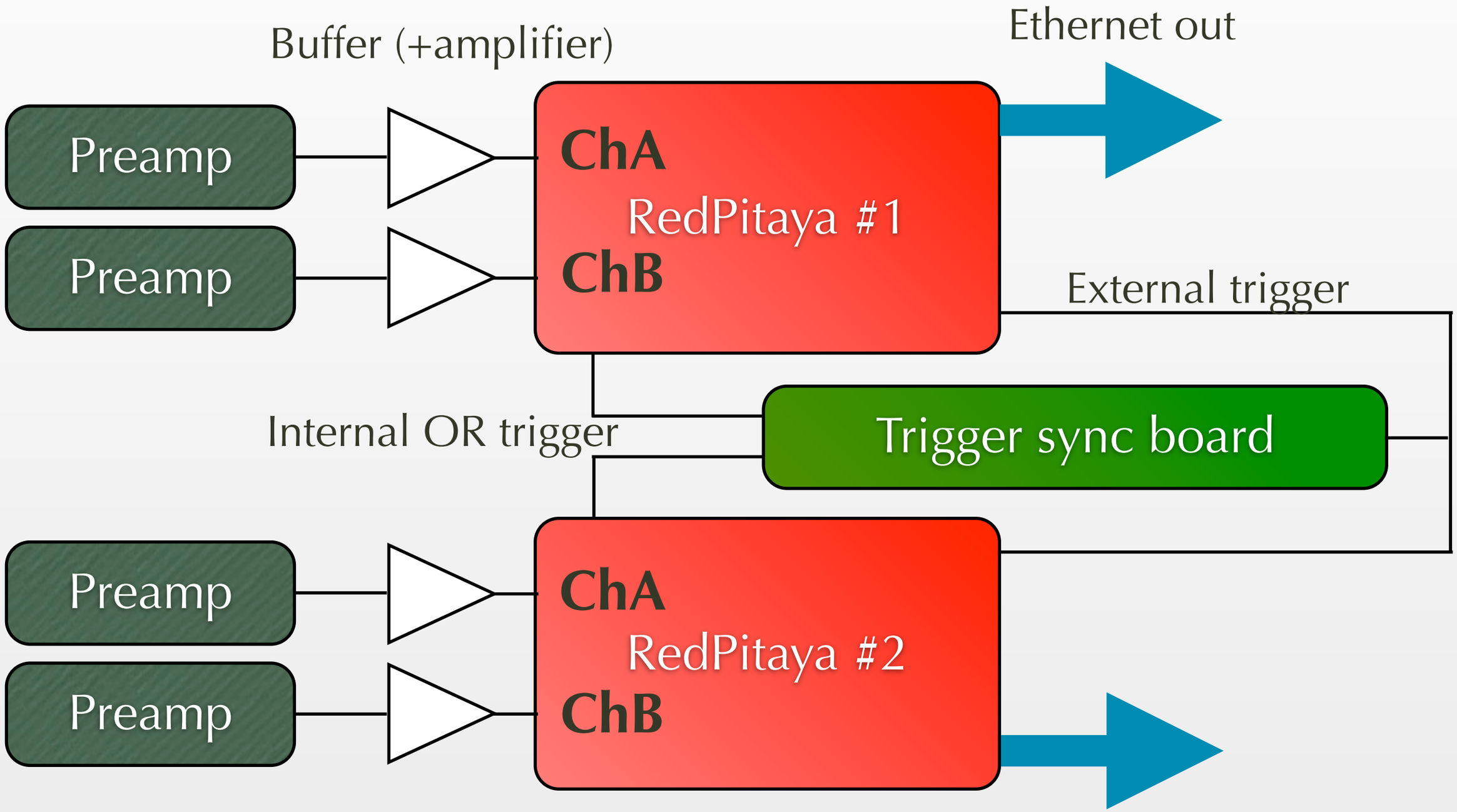
First prototype with 2 boards done,
with customized FPGA bitfile

Client (C++) + Server
(Python) interface programs
Connectivity via Ethernet
and socket

Main customizations with
respect to the default bitfile:
1. Any decimation sampling
now available
2. Internal OR trigger
3. Triple OR trigger (int+ext)

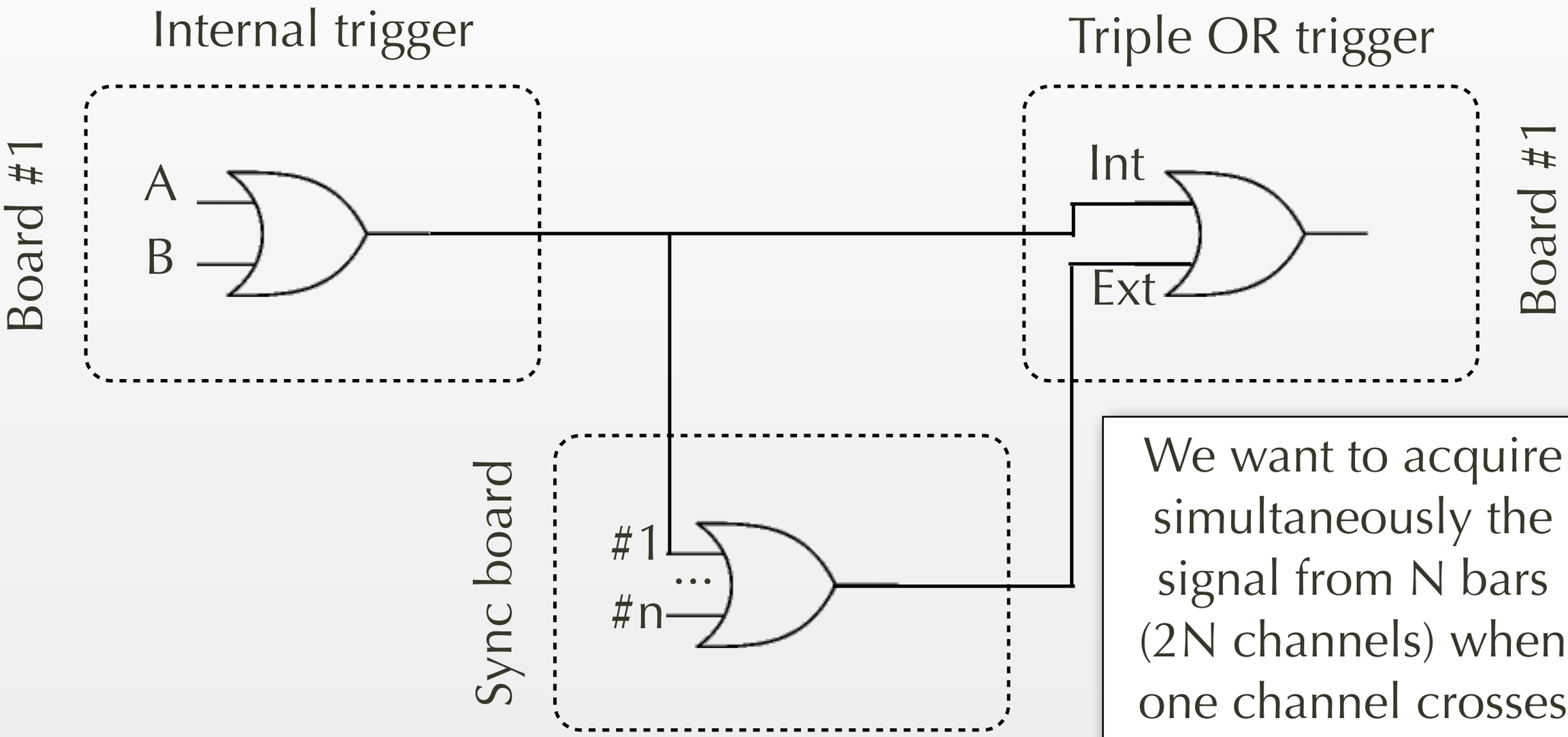


DSP with RedPitaya



DSP with RedPitaya

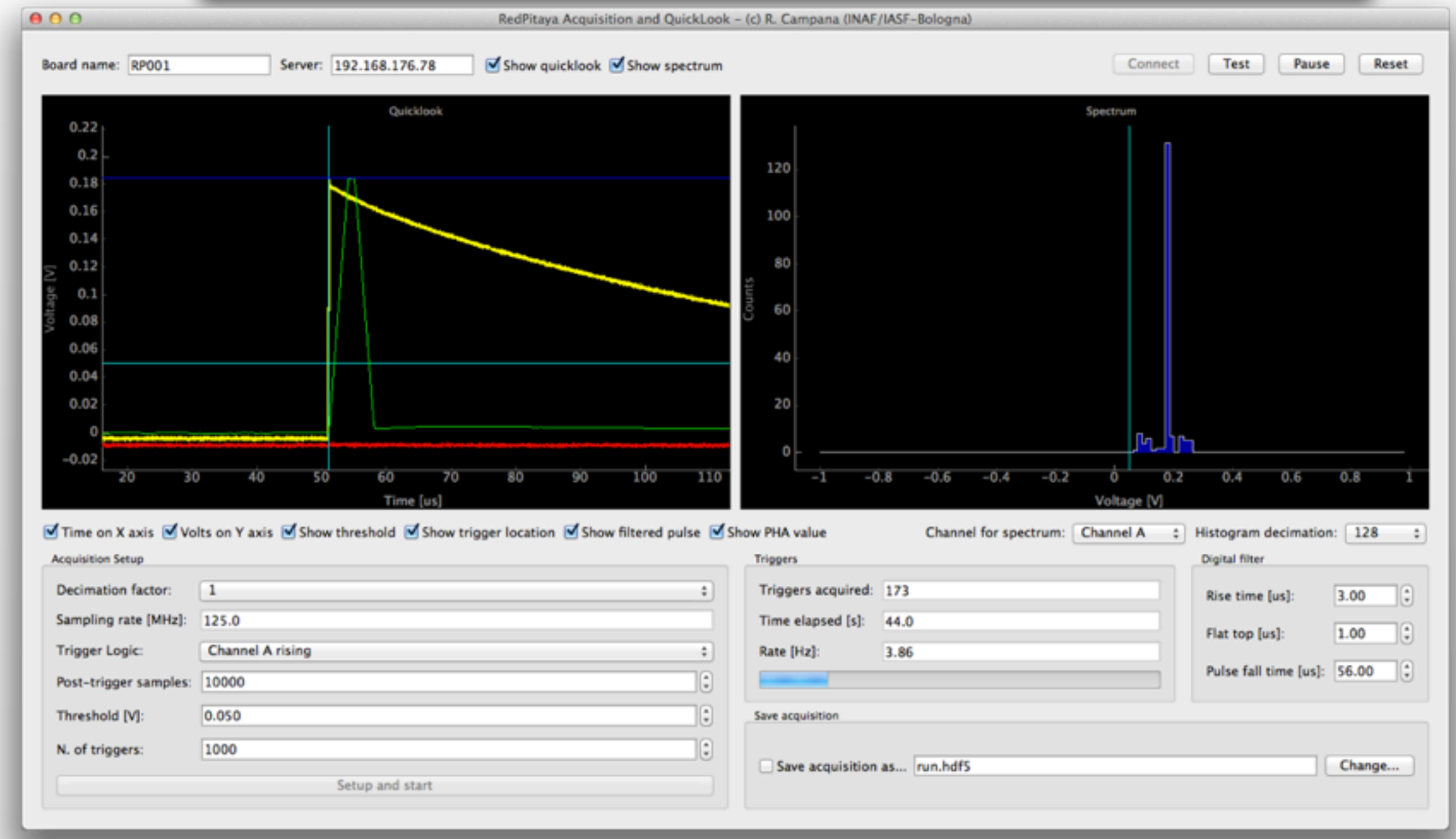
Trigger logic



We want to acquire simultaneously the signal from N bars (2N channels) when one channel crosses the threshold

DSP with RedPitaya

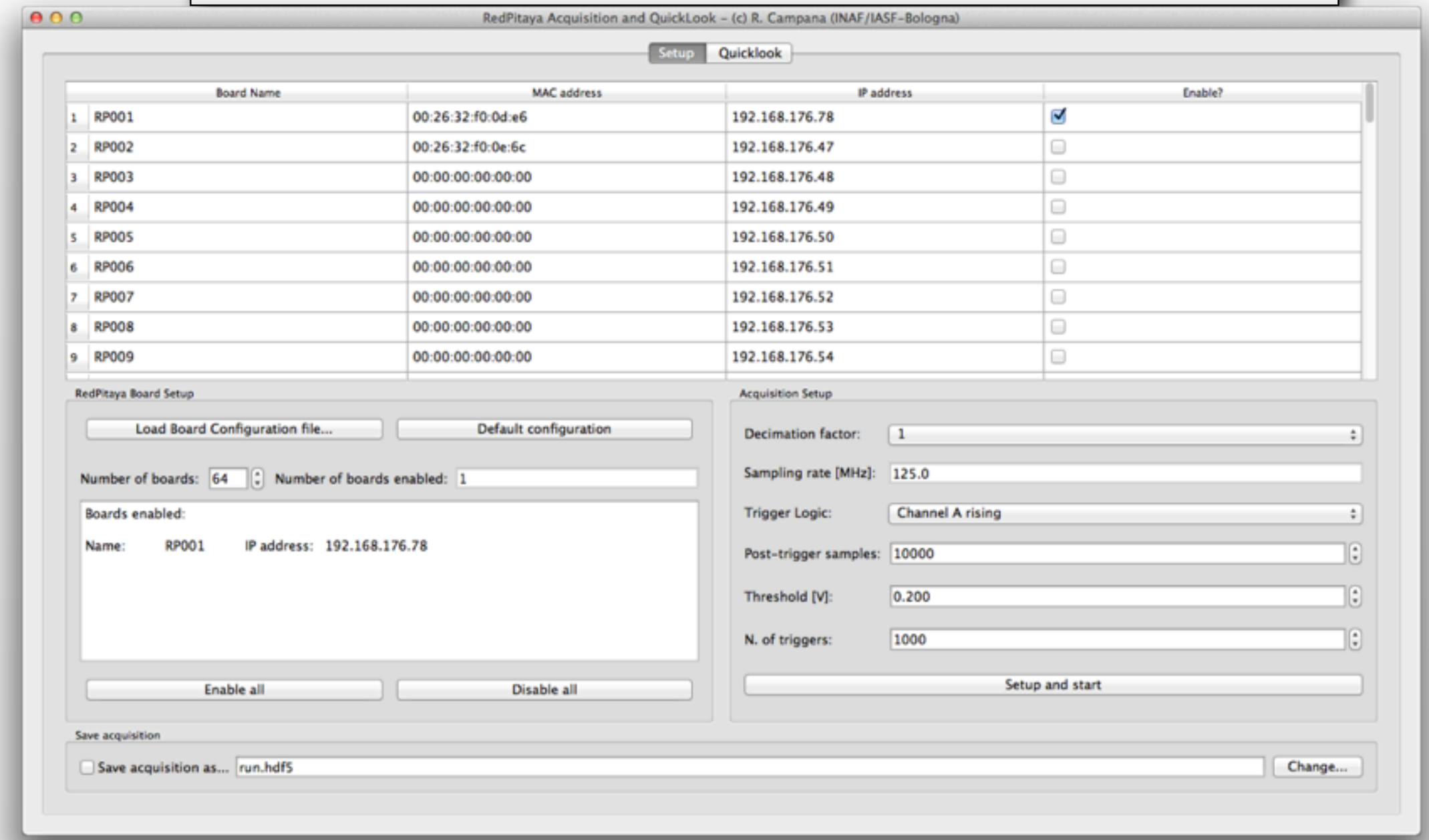
Successfully implemented a multi-platform GUI test equipment interface (python + QT4)



Single board version

DSP with RedPitaya

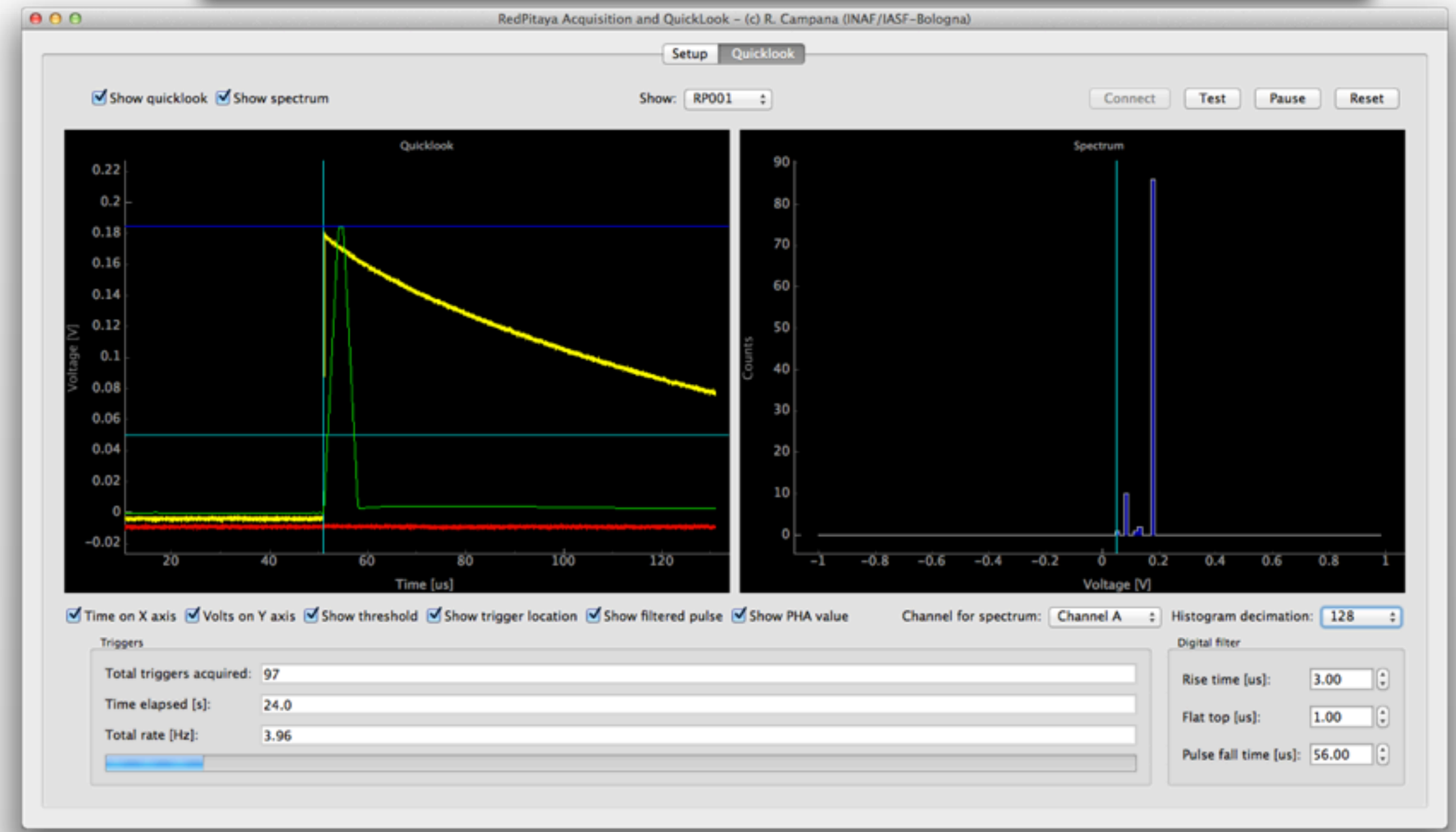
Successfully implemented a multi-platform GUI test equipment interface (python + QT4)



Multiple boards version

DSP with RedPitaya

Successfully implemented a multi-platform GUI test equipment interface (python + QT4)



Multiple boards version

Future work

1. Implement a N-boards scalable system (almost done)
2. Complete acquisition software (almost done)
3. Integration of a 16 bars (32 channels) module
4. Integration of up to 4 modules (64 bars, 128 ch.)
5. Testing and characterization of the architecture

Thanks!

Backup slides

Silicon Drift Detectors