

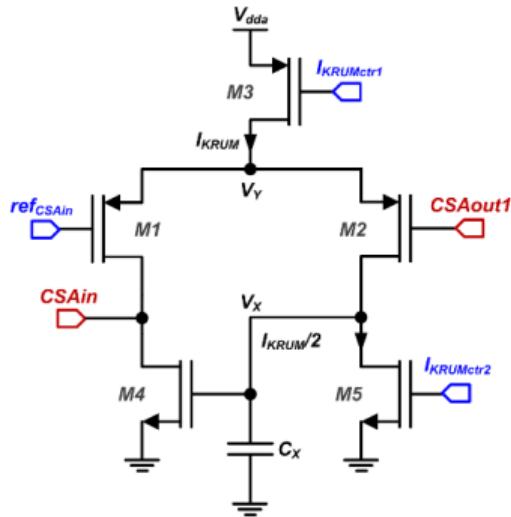


Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Annika Vauth
17 August 2021

CSADigitizer

A Krummenacher Digitizer Module for ap²

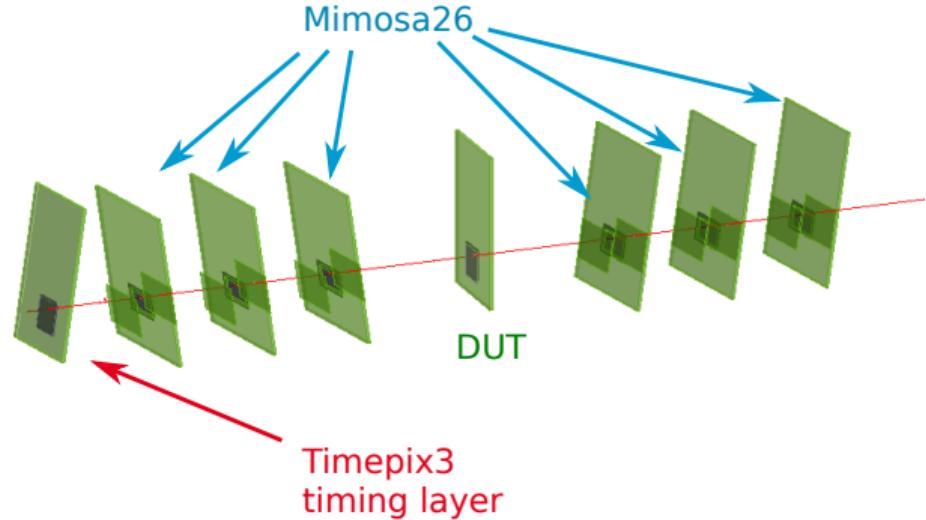


Motivation

Planned upgrade for DESY
test beam telescopes:
timing layer

Test a few things in simulation

Tool of choice: Allpix²



Allpix² simulation chain



Allpix² simulation chain

- Geometry Construction,
- Electric Field Configuration
- Energy Deposition
- Charge Transport
- Signal Transfer
- Digitization: no timing capabilities in DefaultDigitizer

Allpix² simulation chain



Allpix² simulation chain

- Geometry Construction,
- Electric Field Configuration
- Energy Deposition
- Charge Transport
- Signal Transfer
- Digitization: no timing capabilities in DefaultDigitizer

Implement a new digitizer:

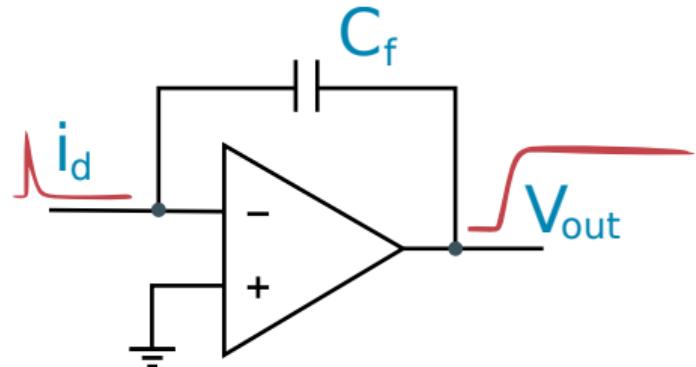
Charge Sensitive Amplifier (CSA) with Krummenacher feedback

→ *CSADigitizer module*

Charge Sensitive Amplifier

Basic operating principle:

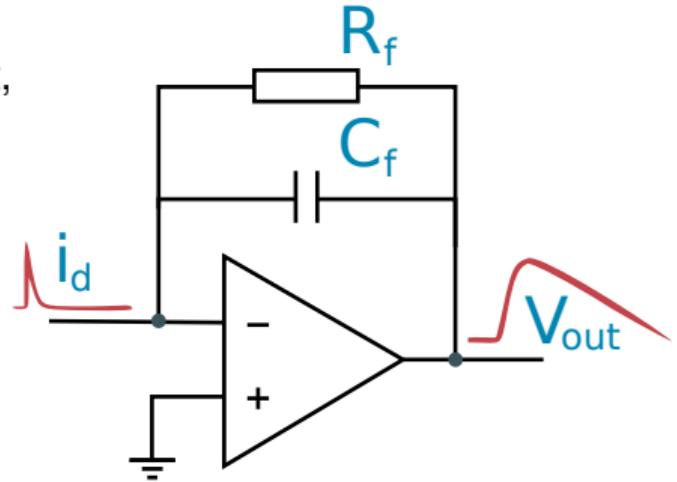
- ▶ feedback capacitor C_f between input and output, stores the charge from the detector
- ▶ current from the detector causes the output to step (integration signal)
- ▶ gain in units of output volts over input charge, scales with $1/C_f$



Charge Sensitive Amplifier

Basic operating principle:

- ▶ feedback capacitor C_f between input and output, stores the charge from the detector
- ▶ current from the detector causes the output to step (integration signal)
- ▶ gain in units of output volts over input charge, scales with $1/C_f$
- ▶ reset the CSA by placing a high-value resistor in parallel with C_f
- ▶ (not drawn) inductive path for leakage current compensation → Krummenacher



Implementation in Allpix²: Transfer function

Idea for CSA implementation:

Transfer function from [Kleczek 2016 JINST11 C12001]

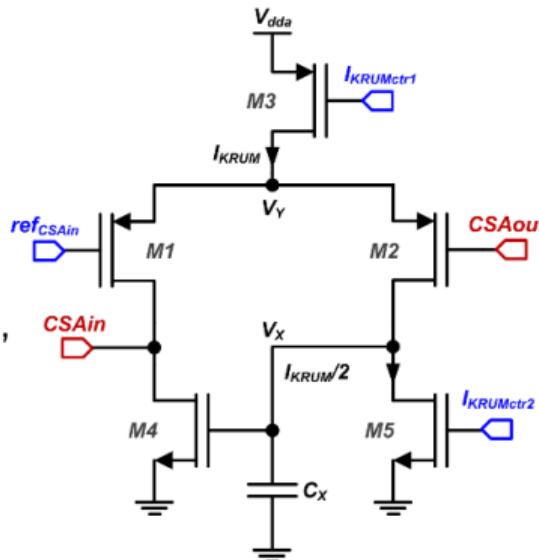
$$H(s) = \frac{R_f}{((1 + \tau_f s) * (1 + \tau_r s))},$$

with $\tau_f = R_f C_f$, rise time constant $\tau_r = \frac{C_{det} * C_{out}}{g_m * C_f}$

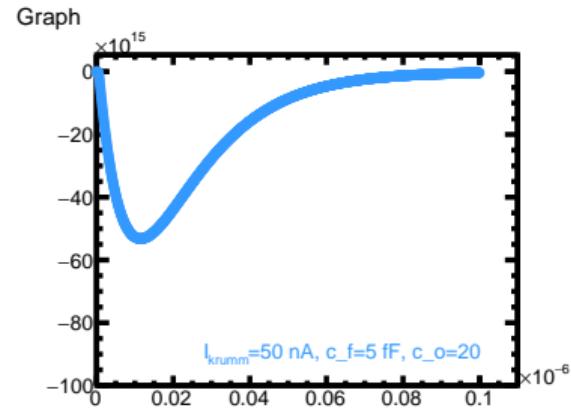
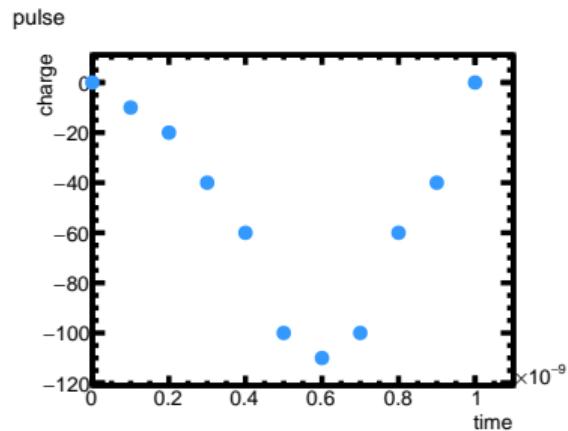
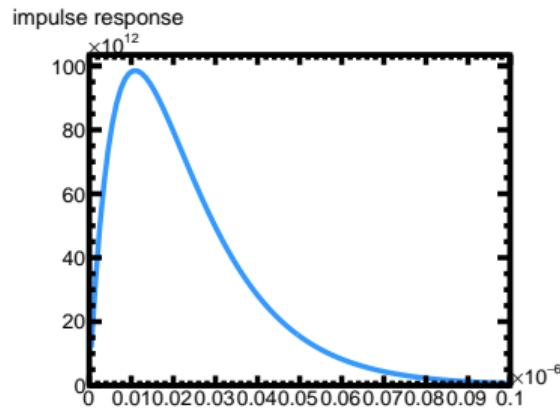
Impulse response: $\mathcal{L}^{-1}(K)$ (thanks to Wolfram Alpha)

$$\text{InverseLaplaceTransform}[R/((1 + as)(1 + sb)), s, t] = (1/((a - b)E^{t/a}) - 1/((a - b)E^{t/b}))R$$

→ convolution of impulse response with charge pulse

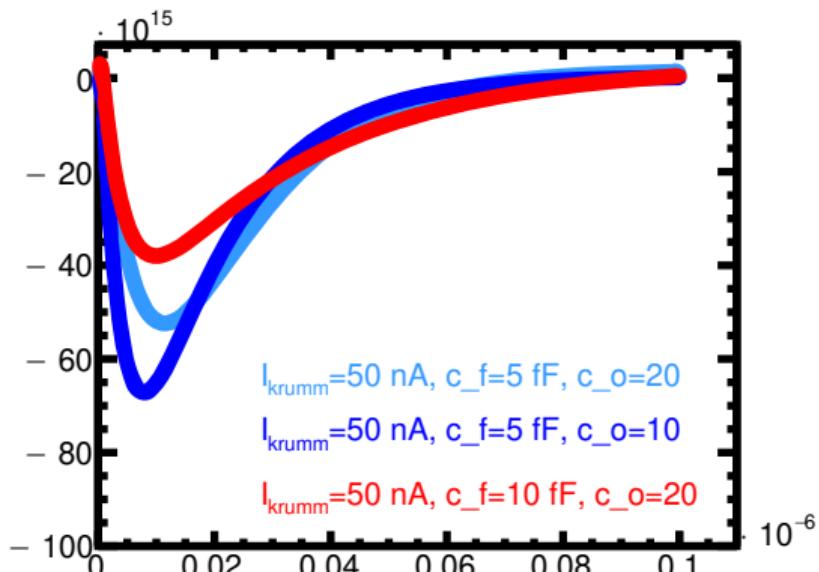
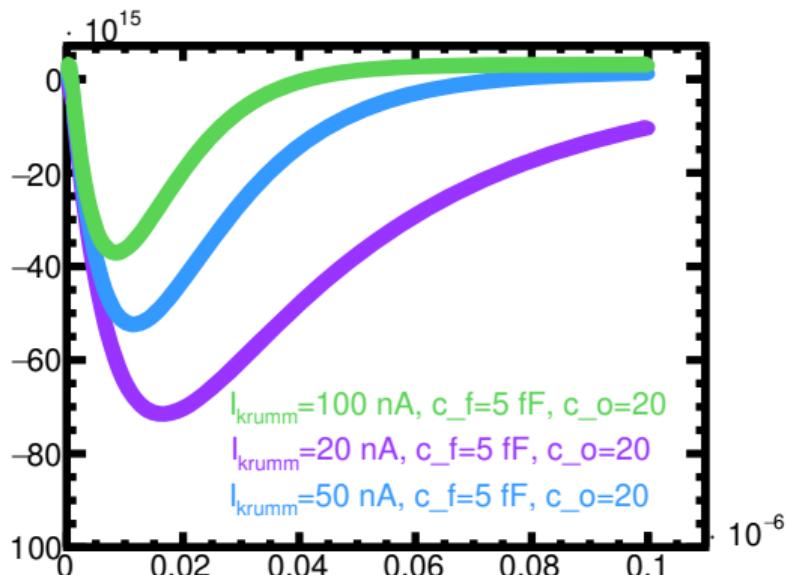


Example with simple dummy pulse



simple toy code for testing:
convolution of impulse response with code

Example with simple dummy pulse

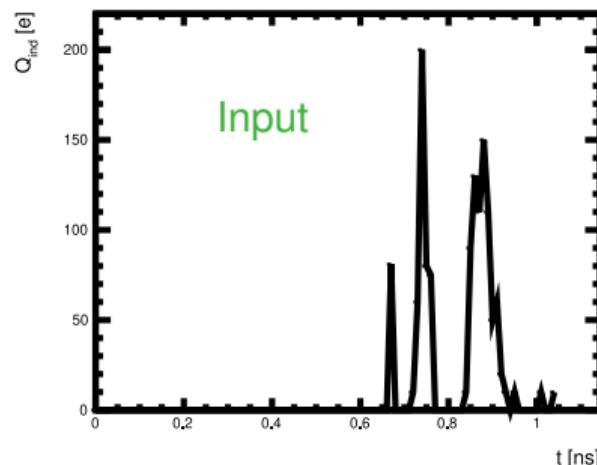


simple toy code for testing:
different I_{Krumm} ; different feedback/output capacities

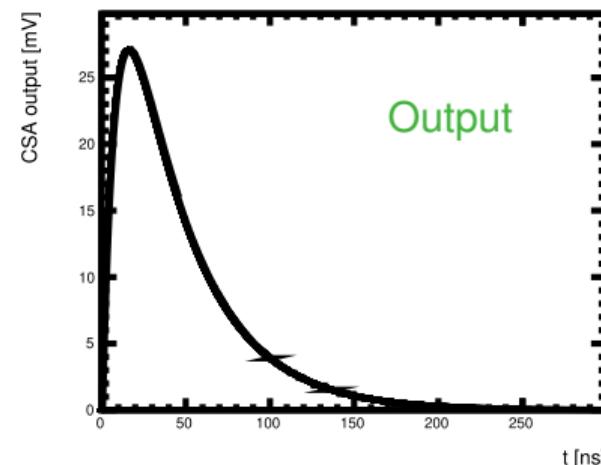
First implementation in Allpix²

- ▶ Convolution of impulse response with charge pulse
→ voltage pulse

Induced charge in pixel (21,22), $Q_{\text{tot}} = 1276 \text{ e}$



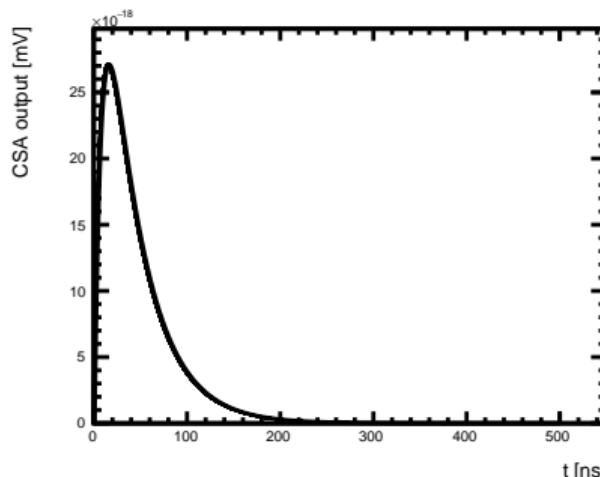
Amplifier signal in pixel (21,22)



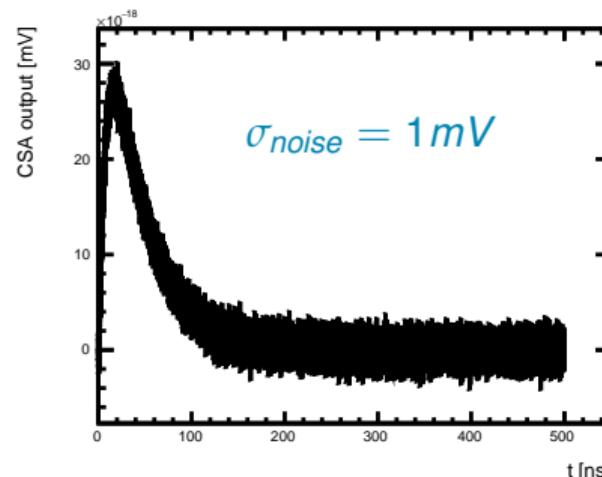
First implementation in Allpix²

- ▶ Convolution of impulse response with charge pulse
→ voltage pulse
- ▶ Addition of noise - simple normal distribution

Amplifier signal in pixel (21,22)



Amplifier signal with added noise in pixel (21,22)



General Parameters

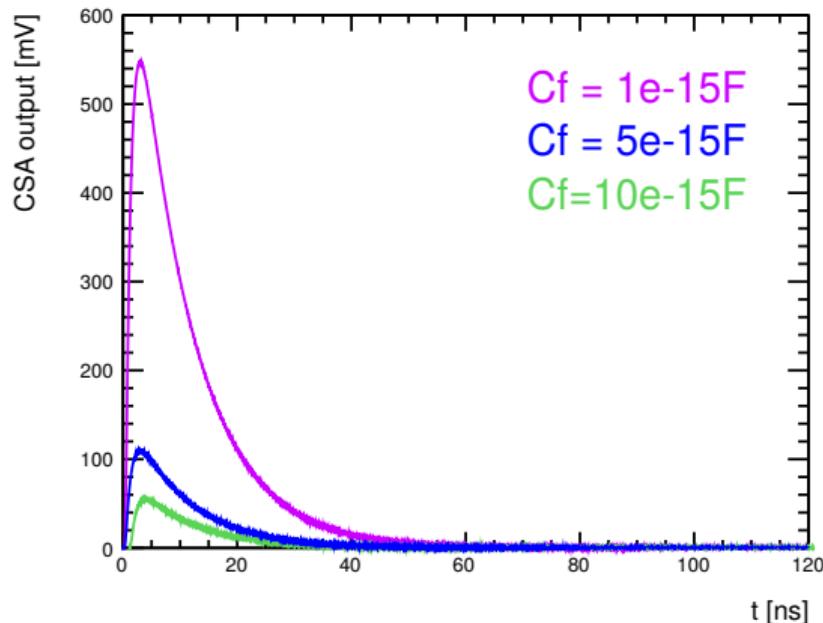
In the first implementation

- ▶ **model** Choice between different CSA models - initially implemented: two parametrisations of the circuit from [Kleczek], ‘simple’ and ‘csa’.
- ▶ **feedback_capacitance** The feedback capacity to the amplifier circuit
- ▶ **integration_time** The length of time the amplifier output is registered
- ▶ **sigma_noise** Standard deviation of the Gaussian-distributed noise added to the output signal
- ▶ **output_plots [output_pulsegaphs]** Enable output histograms: raw pixel charge, time of arrival, time over threshold, pixel charge vs tot *[pulse graphs for each pixel and event]*.

Example output: C_f

Example pulse shape for different feedback capacitors (1 fF , 5 fF, 10 fF)

Amplifier signal with added noise in pixel (755-366)



one sensor, otherwise default parameters

Parameters for ToA/ToT logic implementation

ToT/ToA logic:

check when pulse crosses threshold

- ▶ **threshold:**

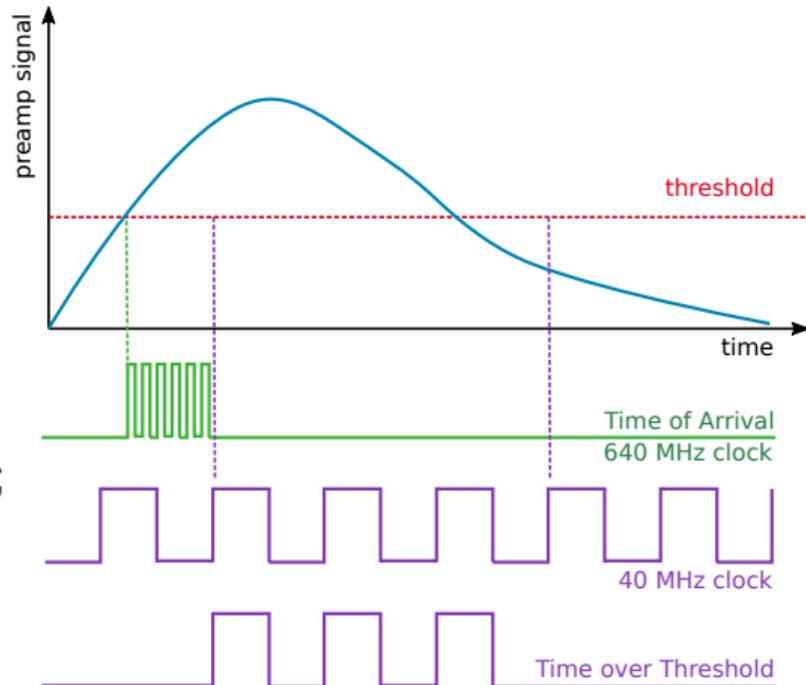
Threshold for TOT/TOA logic,
for considering the output signal as a hit

- ▶ **clock_bin_tot:**

Xcycle for time-over-threshold clock
(if set, output signal is ToT in clock cycles;
if not set, pulse integral is returned)

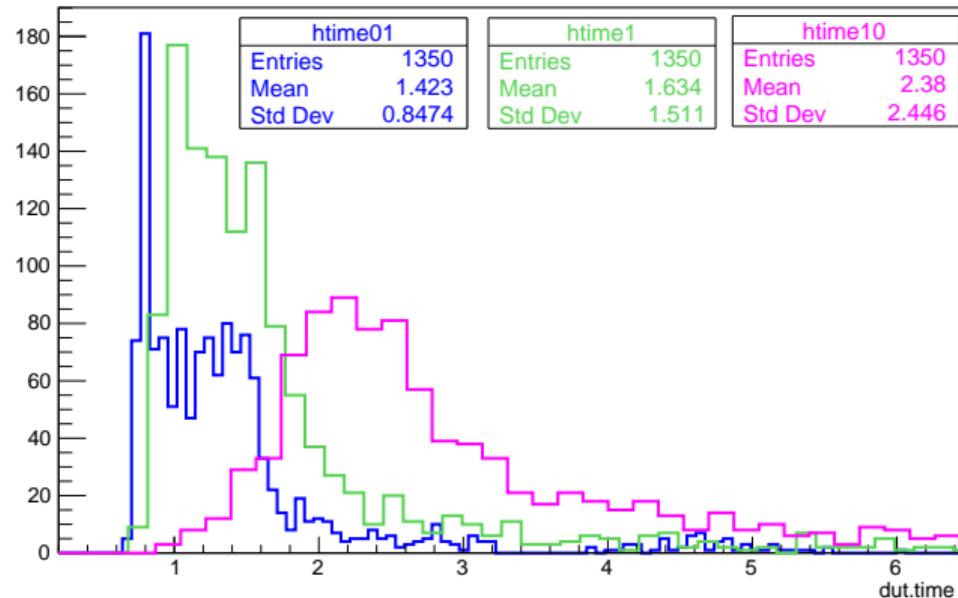
- ▶ **clock_bin_toa:**

Xcycle for the time-of-arrival clock
(if not set, output time is in ns instead)



Test: Thresholds/ToA

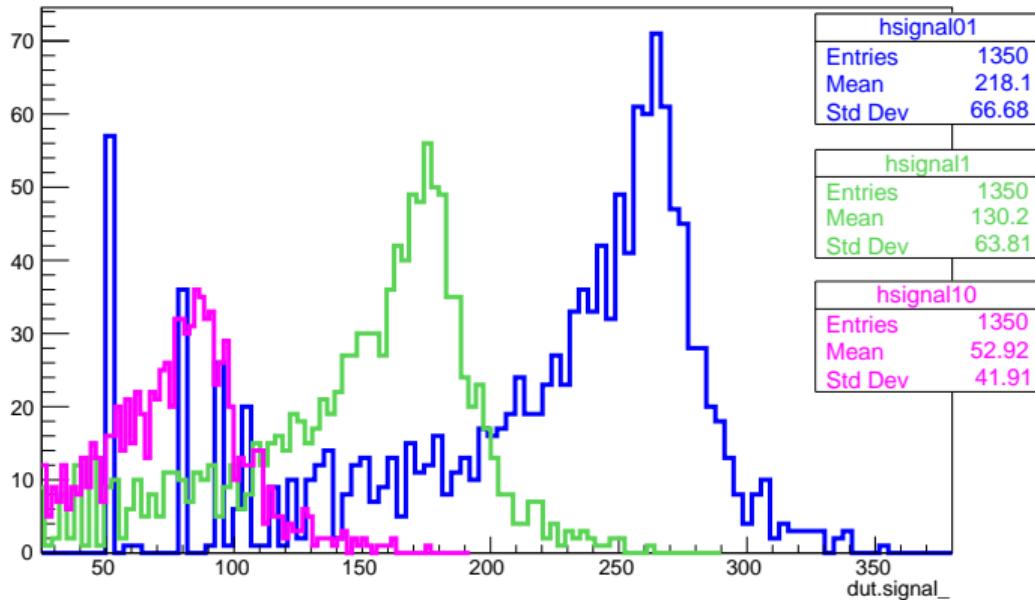
Time-of-Arrival for different thresholds (0.1mV , 1mV , 10mV)
dut.time_ different thresholds



one sensor, otherwise default parameters

Test: Thresholds/ToT

Time-over-Threshold for different thresholds (0.1mV , 1mV , 10mV)
dut.signal_ different thresholds



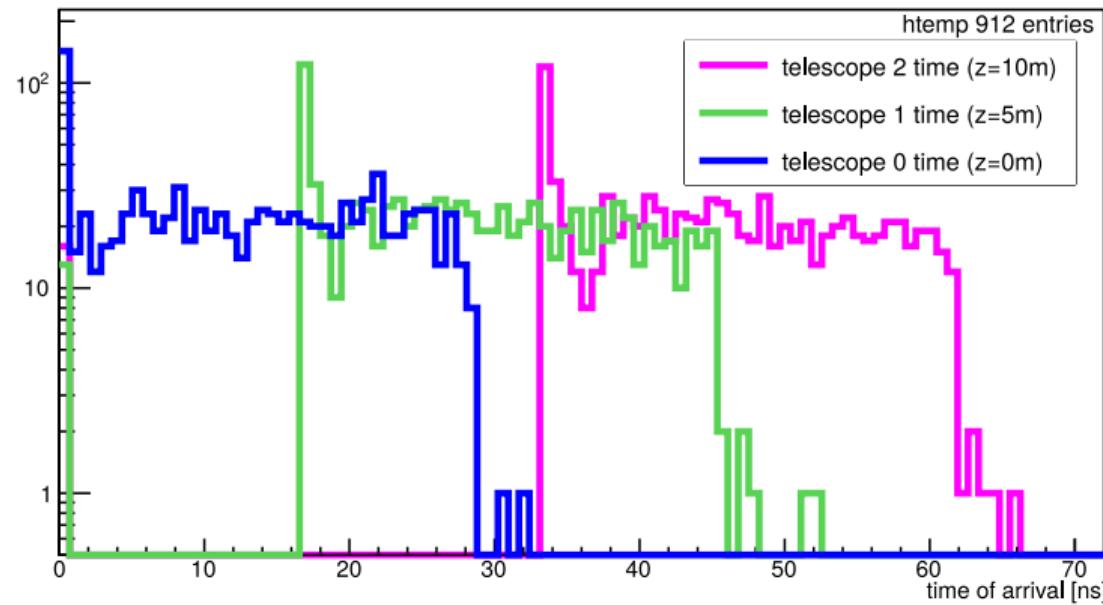
one sensor, otherwise default parameters

Test: Timing of the setup

Time-of-Arrival for **very** exaggerated dummy setup:

3 detector planes, spaced 5 meter apart

TOA for three detector planes



Specific Parameters

Parameters for the `csa` model:

- ▶ **krummenacher_current** The feedback current setting of the CSA.
- ▶ **detector_capacitance** The detector capacitance.
- ▶ **amp_output_capacitance** The capacitance at the amplifier output.
- ▶ **transconductance** The transconductance of the CSA feedback circuit.
- ▶ **temperature** Defaults to 293.15K.

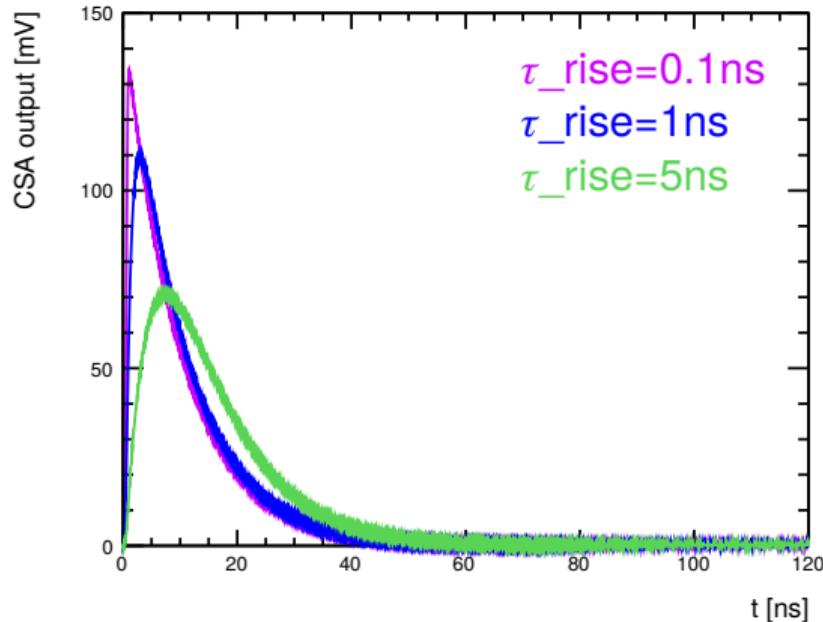
Parameters for the `simple` model:

- ▶ **rise_time_constant** Rise time constant of CSA output.
- ▶ **feedback_time_constant** Feedback time constant of CSA output.

Example output: τ_{rise}

Example pulse shape for different rise times (0.1 ns, 1 ns, 5 ns)

Amplifier signal with added noise in pixel (755-366)

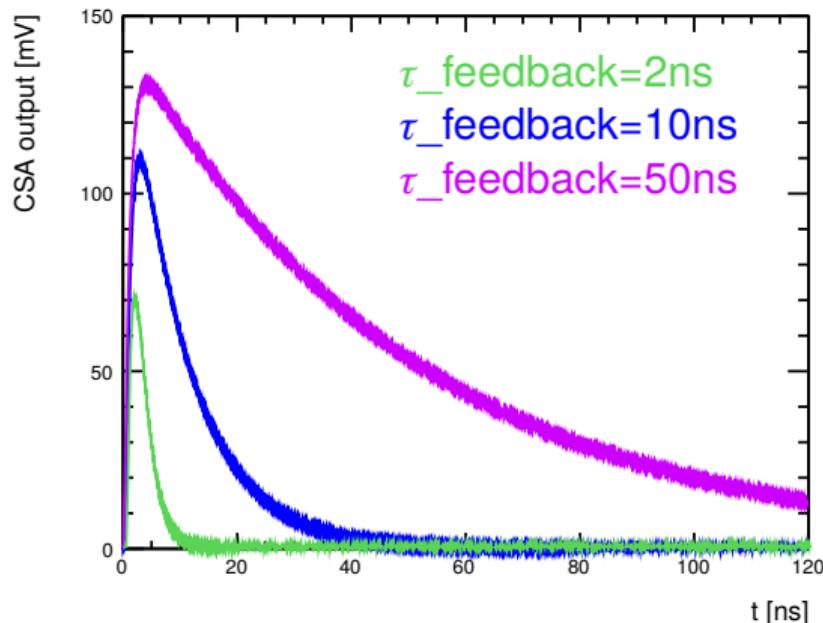


one sensor, otherwise default parameters

Example output: τ_{feedback}

Example pulse shape for different feedback times (2 ns, 10 ns, 50 ns)

Amplifier signal with added noise in pixel (755-366)



one sensor, otherwise default parameters

CSADigitizer Module: Status

Module part of Allpix² v1.5.1 (~ one year ago)

Some of the improvements since then:

[MR 358&360](#) Tests, clarifications, polarity

(add response function graph to modules file,
make sure Krummenacher feedback is positive definite,
properly treat polarity of charge ...)

[MR 461](#) Catch empty pulse in CSA Digitizer

(Note: CSADigitizer requires pulses to work
→ use with PulseTransfer module)

[MR 455](#) Variable CSA response curve

impulse response functions can be provided as ROOT::TFormula → custom model

16 Sep, 2020 1 commit



CSADigitizer: allow processing of positive and negative thresholds
Simon Spannagel authored 10 months ago

19 Apr, 2021 1 commit



Throw exception if no pulse is available for CSA Digitizer
Paul Schuetze authored 3 months ago

08 Apr, 2021 3 commits



CSADigitizer: remove unused class members
Stephan Lachnit authored 4 months ago



CSADigitizer: add custom model using ROOT::TF1
Stephan Lachnit authored 4 months ago

More suggestions very welcome - looking forward to hear about applications!